

Table of Contents

Part I Document Overview	2
Part II Document Details	3
Part III Setup	4
1 Download & Installation	4
2 Security for root	5
3 Data Storage	6
Part IV Setting up a Database, Table and Users	8
1 MySQL SQL Shell	8
2 MySQL Control Center	10
Setting up the Control Center	10
Creating the Database and Table	13
Creating Users	15
Part V Appendix	19
1 Security Tips	19
2 ODBC Drivers	20
3 Apache & PHP	21
Index	0

1 Document Overview



Author: NETIKUS.NET ltd
Date: 6th February 2004
Revision: 1.0

MySQL Server Installation on Windows (for EventSentry)

Title	MySQL Server Installation on Windows
Summary	How to install, setup and configure a MySQL installation on Windows. Includes instructions on how to setup a database for EventSentry, also covers the installation of Apache and PHP briefly.
Software	MS Windows 2000 or higher MySQL 4.0 Optional: Full or trial version of EventSentry
Hardware	Not applicable, system used for this guide is PIII 977MHz with 256Mb RAM
Skill Level	Beginner - Intermediate
Skills Required	- Basic understanding of Windows NT, 2000, 2003 or XP - Basic understanding of databases - Knowledge of basic SQL commands (select, update, ...) beneficial but not required
Acknowledgements	Thanks to the MySQL database team for making a great database available for free
Download	http://www.netikus.net/ (guides section)

2 Document Details

Overview

This document describes how to install a MySQL database server on Windows 2000, mainly focusing on setting up the server with the EventSentry database and table.

The appendix also gives instructions on how to install Apache and PHP.

This guide is also useful if you are not using EventSentry as it covers mostly basic MySQL tasks.

MySQL

MySQL is an open source relational database and can hence be used for free. It lacks some features of commercial database, but is an excellent solution for many needs, including EventSentry.

One of the downsides to MySQL is its lack of graphical administration interfaces, though this seems to be improving over time. We found the MySQL Control Center to be quite capable and easy to use, though not as appealing as other products including the MS Sql Enterprise manager.

Why?

This guide was written to help EventSentry customers setup a reliable and affordable database server to store and consolidate event log records.

3 Setup

3.1 Download & Installation

Download

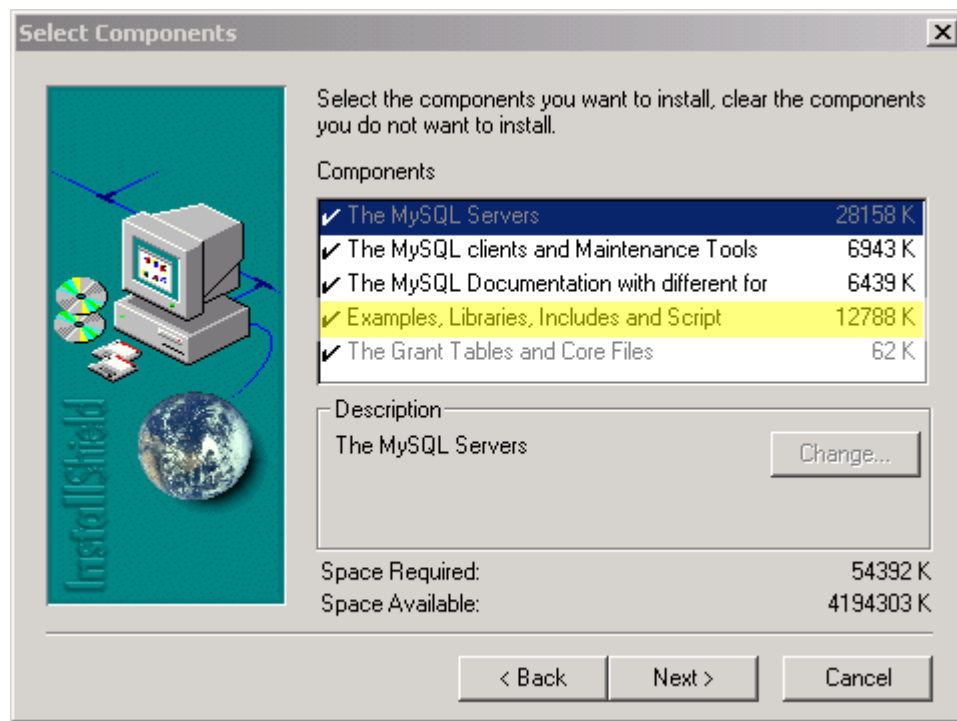
You can download the MySQL database from the MySQL website <http://www.mysql.com> by clicking on the [downloads](#) tab. Scroll down to the *MySQL database server & standard clients* section and select the latest production release of MySQL, 4.0 at the time of writing.

After clicking on the link you will be taken to the download page, scroll down to locate the *Windows Downloads* section and choose the first *Pick A Mirror* link. Now select a mirror and download the archive. Note that you will need [Winzip](#) or a similar utility to unzip the file you just downloaded.

We also recommend downloading the [MySQL Control Center](#) from the downloads page. It's currently in beta stage but allows you to administer your MySQL server with a graphical user interface. We will cover both ways of administration in this guide.

Installation of MySQL Server

Unzip the setup file and execute **setup.exe**. You can just keep all the default options during the installation, but you may skip the examples, libraries etc:



MySQL installs into `c:\mysql` by default which is fine for most cases, we will be referencing to this directory in this guide. Please note that the installation does not setup MySQL to run as a service by default.

Creating the MySQL Service

In almost all cases you will want to run the MySQL database as a Windows service so that it is always

active and started automatically when the server is started. To install the service open a command prompt and type the following commands:

```
cd c:\mysql\bin
mysqld-nt --install
net start mysql
```

and you should see

```
The MySQL service is starting.
The MySQL service was started successfully.
```

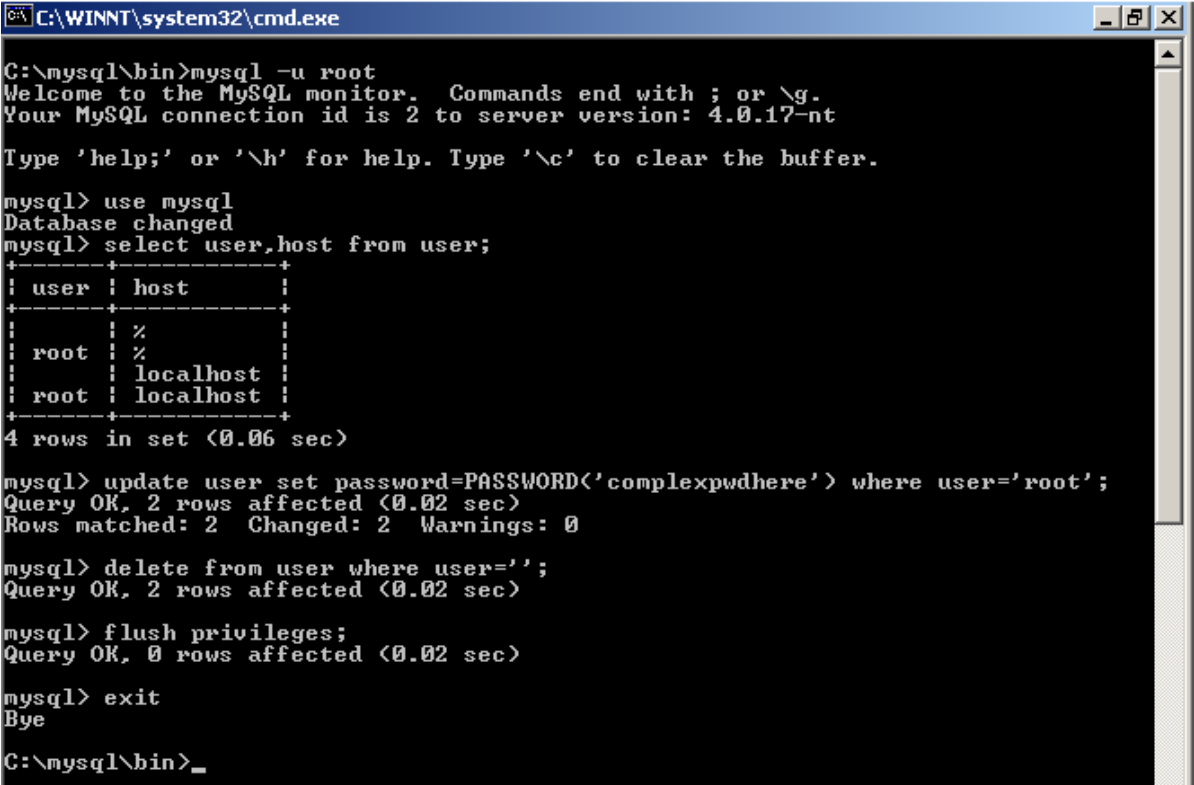
Now the MySQL service is installed and running and we can proceed. If you are monitoring your services with EventSentry then you should now be notified that a new service **MySQL** was added.

Installation of the MySQL Control Center

Simply execute setup.exe and follow the instructions on the screen. The application uses little disk space and a restart is not required after the installation. Once installed you can launch the application by double-clicking the MySQL Control Center icon on the desktop.

3.2 Security for root

The root user does not have a password by default which is a security risk and needs to be changed immediately. We will change the root password from the command line using the mysql SQL shell. Open the command prompt and follow the instructions in the screenshot:



```
C:\WINNT\system32\cmd.exe
C:\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 4.0.17-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql
Database changed
mysql> select user,host from user;
+-----+-----+
| user | host |
+-----+-----+
| root | %    |
| root | %    |
| root | localhost |
| root | localhost |
+-----+-----+
4 rows in set (0.06 sec)

mysql> update user set password=PASSWORD('complexpwdhere') where user='root';
Query OK, 2 rows affected (0.02 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> delete from user where user='';
Query OK, 2 rows affected (0.02 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.02 sec)

mysql> exit
Bye
C:\mysql\bin>_
```

- First we connect to the SQL instance as the **root** user by using the **-u** option. We do not have to specify a password since the root user does not have a password by default.

- Then we change the database to the built-in **mysql** database, that contains all usernames among other things
- The select command here shows that four user accounts exist by default, two for the user root and two anonymous ones.
- We change the password of both root user accounts (explanation for both is below) by using a standard SQL update command. The **PASSWORD()** function is used to create the password hash, just specifying the password in quotes would not work.
- We then remove the anonymous user account with a delete statement.
- Changes to the root user account password do not always seem to become effective immediately, so we force this internal update by issuing the **flush privileges** command.

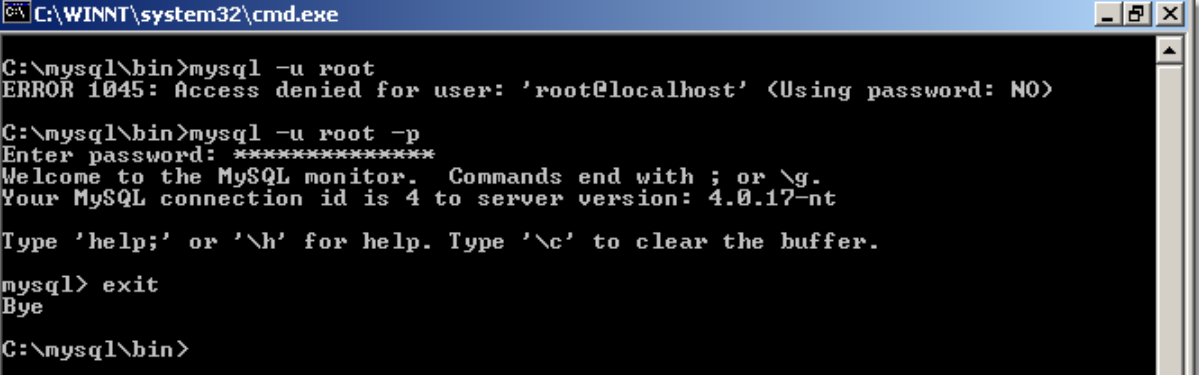
After typing exit we have successfully remove two unneeded user accounts and set a password for the root user.

Unlike other database servers, MySQL allows you to link a database user with a hostname. This allows you to create different users (and hence different permissions) depending on where a user is connecting from.

For example, by default there are two root accounts with different **host** values. One account has the host value set to **localhost**, whereas the other has it set to **%** - a wildcard matching all host names.

This feature is not relevant to us right now since we set the same password for both root accounts, but it's important and helpful to know that this feature exists. When you connect to a MySQL database then the server will always perform a reverse lookup to match the correct username.

From now on we can no longer just type `mysql -u root`, but instead will need to specify the password we have just set above by typing `mysql -u root -p`. The `-p` switch tells the application to prompt us for the password.



```
C:\WINNT\system32\cmd.exe
C:\mysql\bin>mysql -u root
ERROR 1045: Access denied for user: 'root@localhost' (Using password: NO)
C:\mysql\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 4.0.17-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> exit
Bye
C:\mysql\bin>
```

3.3 Data Storage

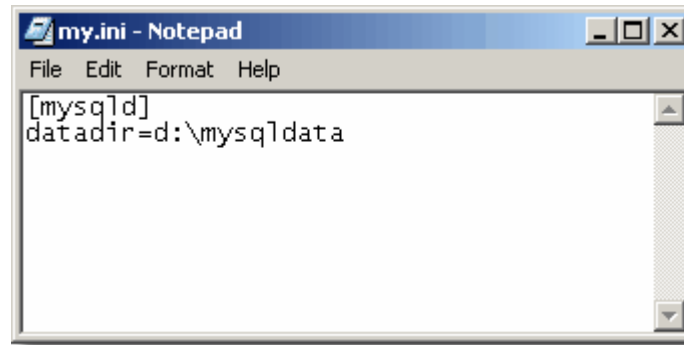
By default MySQL stores all databases in the `c:\mysql\data` directory, however in many cases you might want to store your databases in a different location / drive. You can change the default location by creating a configuration file and changing the **datadir** variable. In this example we will change the datafile director from `c:\mysql\data` to `d:\mysqldata`.

Open a text editor (e.g. notepad) and create the file `%systemroot%\my.ini`. This file should have

the lines

```
[mysqld]  
datadir=d:\mysql\data
```

as shown in the screenshot below:



and save this file as `%systemroot%\my.ini`. Before we restart the MySQL service however we will need to move the contents of the current directory to the new directory, otherwise the crucial built-in **mysql** database will be missing and the MySQL service will not start.

1. Stop the mysql service
2. Create the directory `d:\mysql\data`
3. Move or copy all files and directories from `c:\mysql\data` to `d:\mysql\data`
4. Start the mysql service

If you can log in through the mysql shell then the change was successful.

4 Setting up a Database, Table and Users

Now that our core MySQL installation is pretty much complete we can create a separate database for EventSentry. You can follow these steps even if you are not planning on using EventSentry to learn the basics of creating a database, tables and users. You can delete this database again after you have completed the tutorial.

You can create the database, table and users either through the command-line mysql shell, or with the MySQL Control Center. For beginners we recommend choosing the SQL shell first and then taking a look at the control center.

In a nutshell we will have to perform the following tasks:

- Create a new database
- Create two user accounts, one to read and one to write to the database
- Create the table

[1. MySQL SQL Shell](#)

[2. MySQL Control Center](#)

4.1 MySQL SQL Shell

start the SQL shell by typing

```
mysql -u root -p
```

Note that you will need to be in the `c:\mysql\bin` directory if you have not added the `c:\mysql\bin` directory to your path. All commands shown in this chapter will need to be typed in the SQL shell.

Creating a New Database

Databases are always created with the **create database** command, so type

```
create database EventSentry;
```

to create the EventSentry database.

Creating a Table for EventSentry

EventSentry comes with the file `eventsentry_odbc.sql` that contains the table definition for EventSentry. To create this table we first connect to the newly created database and then execute the SQL file to create the table:

```
use EventSentry;
source c:\program files\eventsentry\odbc_samples\eventsentry_odbc.sql
```

You can always check the columns of a table by issuing the **desc** command followed by the table name. To view the table we just created type

```
desc EventSentry
```

and you should see this output:

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
```

eventnumber	int(11)	YES	NULL
eventlog	varchar(64)	YES	NULL
eventtype	varchar(32)	YES	NULL
eventsource	varchar(128)	YES	NULL
eventcategory	varchar(128)	YES	NULL
eventid	int(11)	YES	NULL
eventuser	varchar(128)	YES	NULL
eventcomputer	varchar(32)	YES	NULL
eventtime	datetime	YES	NULL
eventmessage	text	YES	NULL

10 rows in set (0.00 sec)

Creating Users for EventSentry

So far we have a new database and table, but the only user authorized to access this database is the user root, the only existing user! This needs to be changed and so we create two new user accounts:

```
eventsentry_web
eventsentry_svc
```

The user **eventsentry_web** can be used by the ASP or PHP pages to connect and query the EventSentry database, while the **eventsentry_svc** user (svc = service) will be used by the various EventSentry services to connect and write event log data to the database. We are using two user accounts since both users will be used in different places (web server, servers) and performing different things. This way an intruder can't write to the database if he obtains the **eventsentry_web** users' password and vice versa.

We can create the users and grant the necessary permissions at the same time with the following commands:

```
grant SELECT on EventSentry.* to eventsentry_web identified by 'jUE42&@de';
grant INSERT on EventSentry.* to eventsentry_svc identified by 'm**E6W2FF';
```

That's it, we created a database, a table and the users to access the database. You can see the whole thing in action below:

```
C:\WINNT\system32\cmd.exe - mysql -u root -p

C:\mysql\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.17-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database EventSentry;
Query OK, 1 row affected (0.00 sec)

mysql> use EventSentry;
Database changed
mysql> source c:\program files\eventsentry\odbc_samples\eventsentry_odbc.sql
Query OK, 0 rows affected (0.02 sec)

mysql> grant SELECT on EventSentry.* to eventsentry_web identified by 'jUE42&@de';
Query OK, 0 rows affected (0.00 sec)

mysql> grant INSERT on EventSentry.* to eventsentry_svc identified by 'm**E6W2FF';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

In this chapter we are assuming that you will be running a web server to query the EventSentry database from your web browser. You do not have to create the **eventsentry_web** user if will not be running a web server.

If you plan on running the web server on the same machine as the database server, then you can make your user account setup even more secure. Instead of typing the line

```
grant SELECT on EventSentry.* to eventsentry_web identified by  
'jUE42&@de';
```

type the line

```
grant SELECT on EventSentry.* to eventsentry_web@localhost identified by  
'jUE42&@de';
```

4.2 MySQL Control Center

The MySQL Control Center is not (yet?) included in the setup package of the server and is still in the beta phase. With the limited testing that we have performed we have not identified any errors or problems so far.

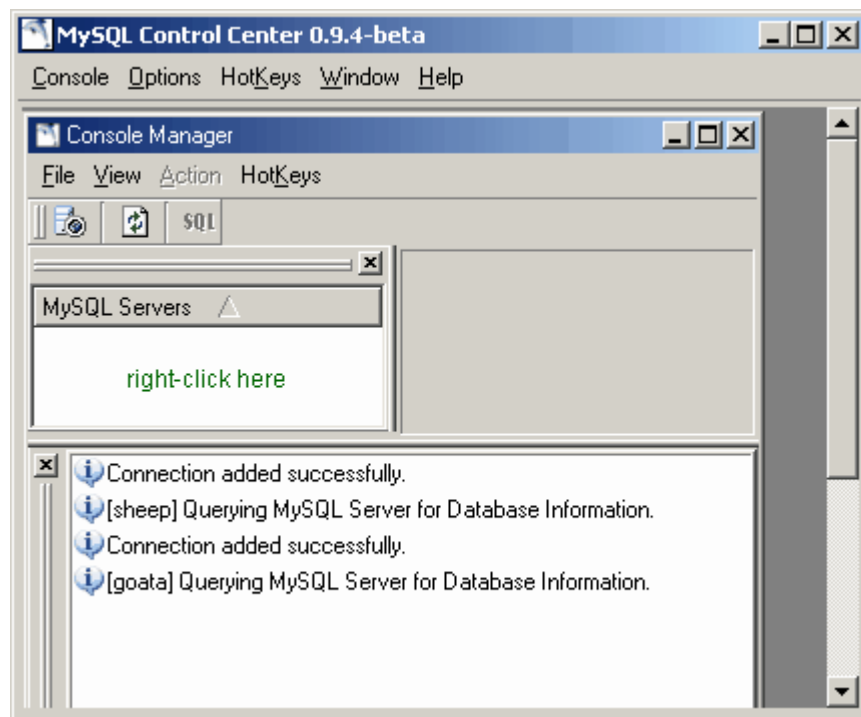
[1. Setting up the Control Center](#)

[2. Creating the Database and Table](#)

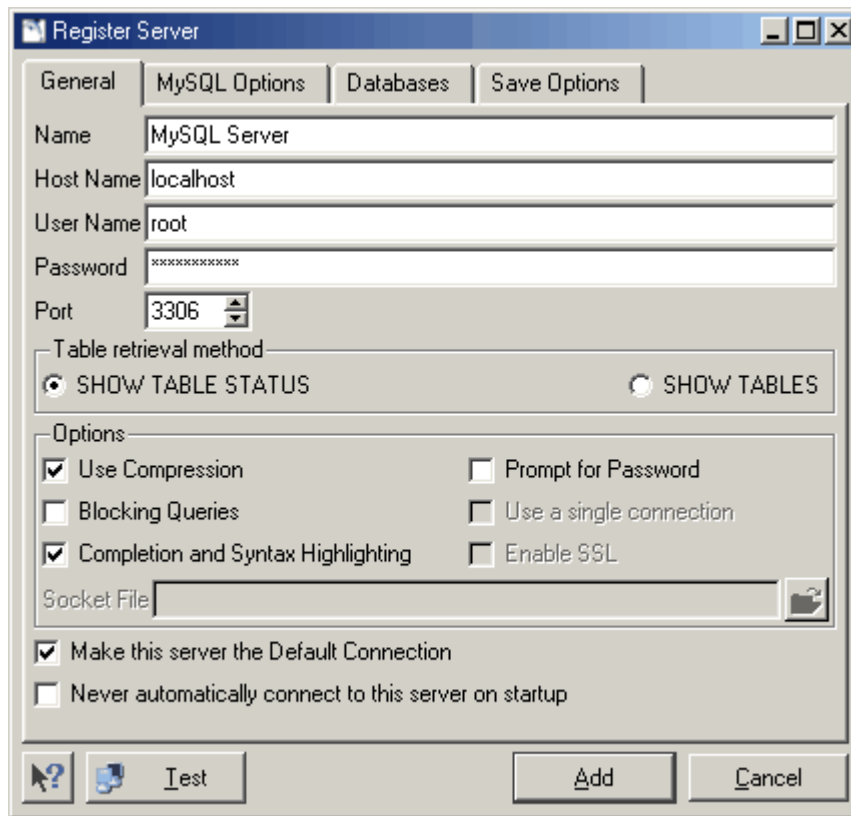
[3. Creating Users](#)

4.2.1 Setting up the Control Center

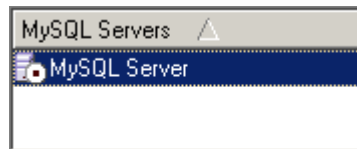
When you start the control center you will be presented with a more or less empty interface



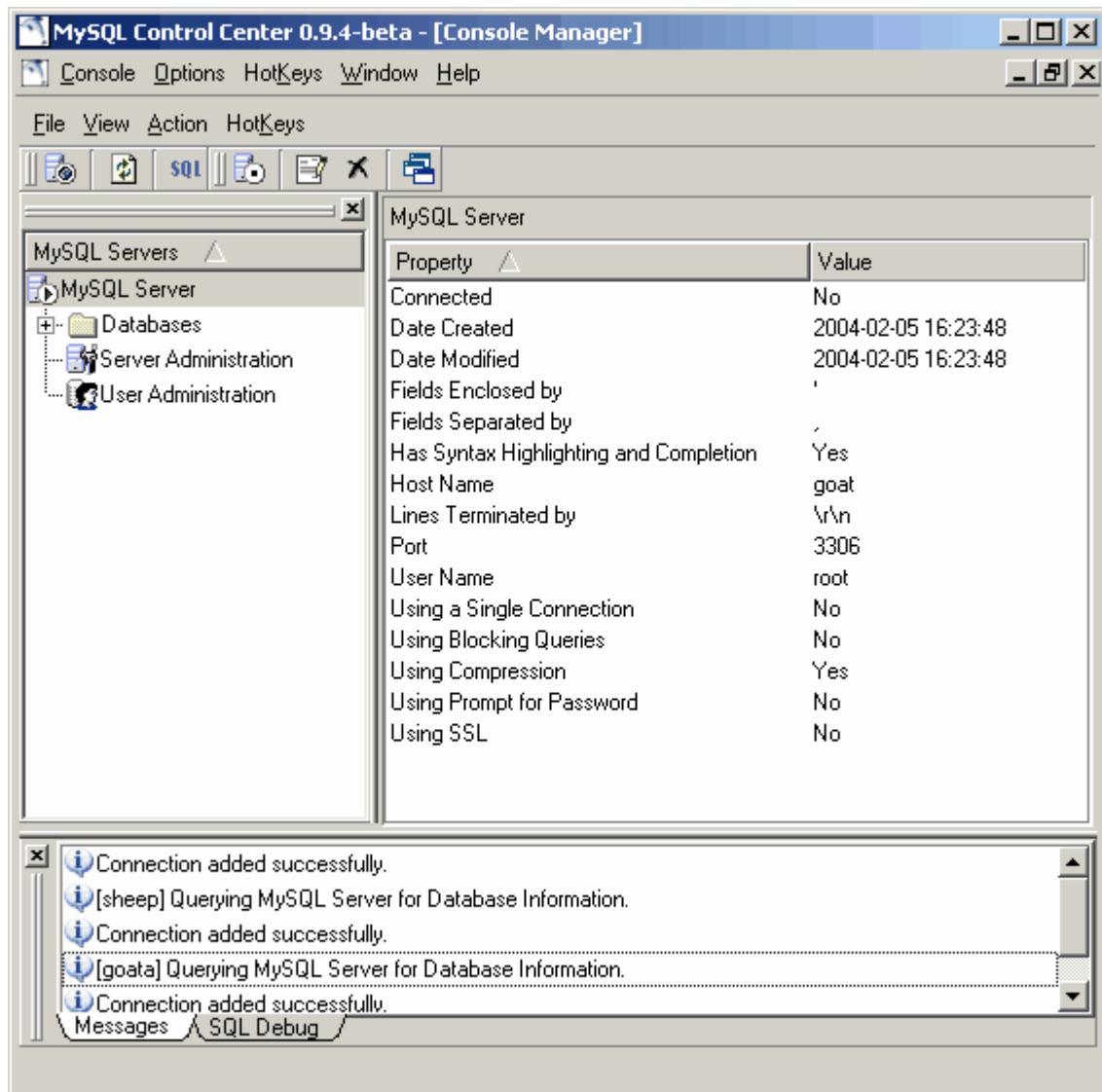
We will need to add and register our MySQL instance in the **MySQL Servers** window so that we can connect and administer our server. Right-click in the white area (see screenshot above) and select **New**. You will be presented with the **Register Server** dialog where you can - register your server! Fill in the first four fields of the dialog as shown below (you may any **Name** to identify this server by) and click the **Add** button once you are done.



The MySQL server should then show up in the server list as shown below.



Now right-click the server and select **Connect**. If you specified the correct information before then you should see a screen similar to this one:

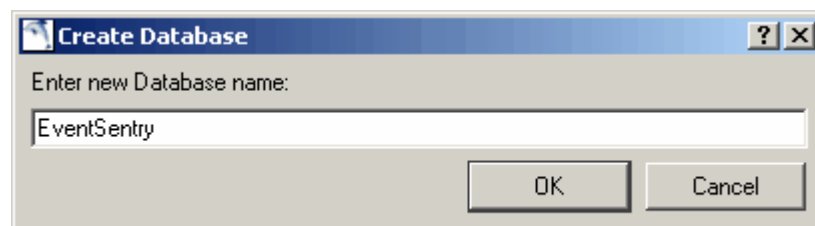


Now that the control center is setup we can continue by creating the database.

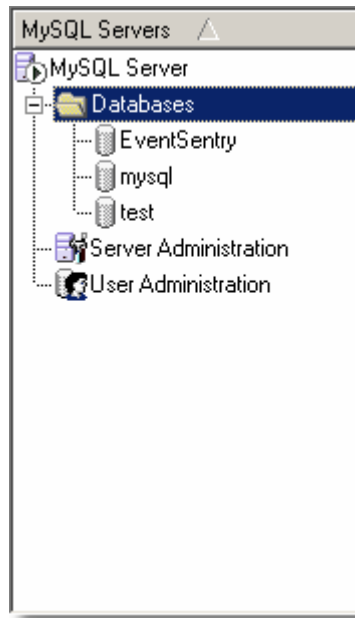
4.2.2 Creating the Database and Table

Creating the Database

Creating the database is quite easy, just right-click the **Databases** container and select **New Database**.

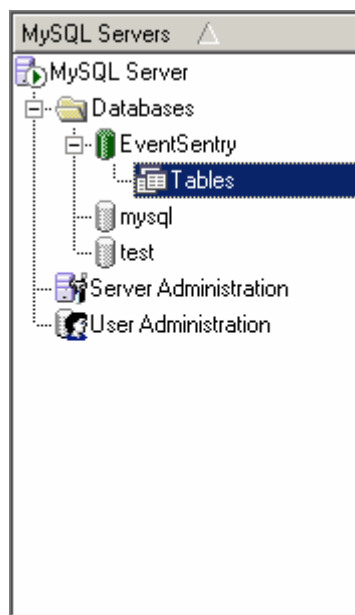


The database should show up in the tree immediately if the database was created successfully:

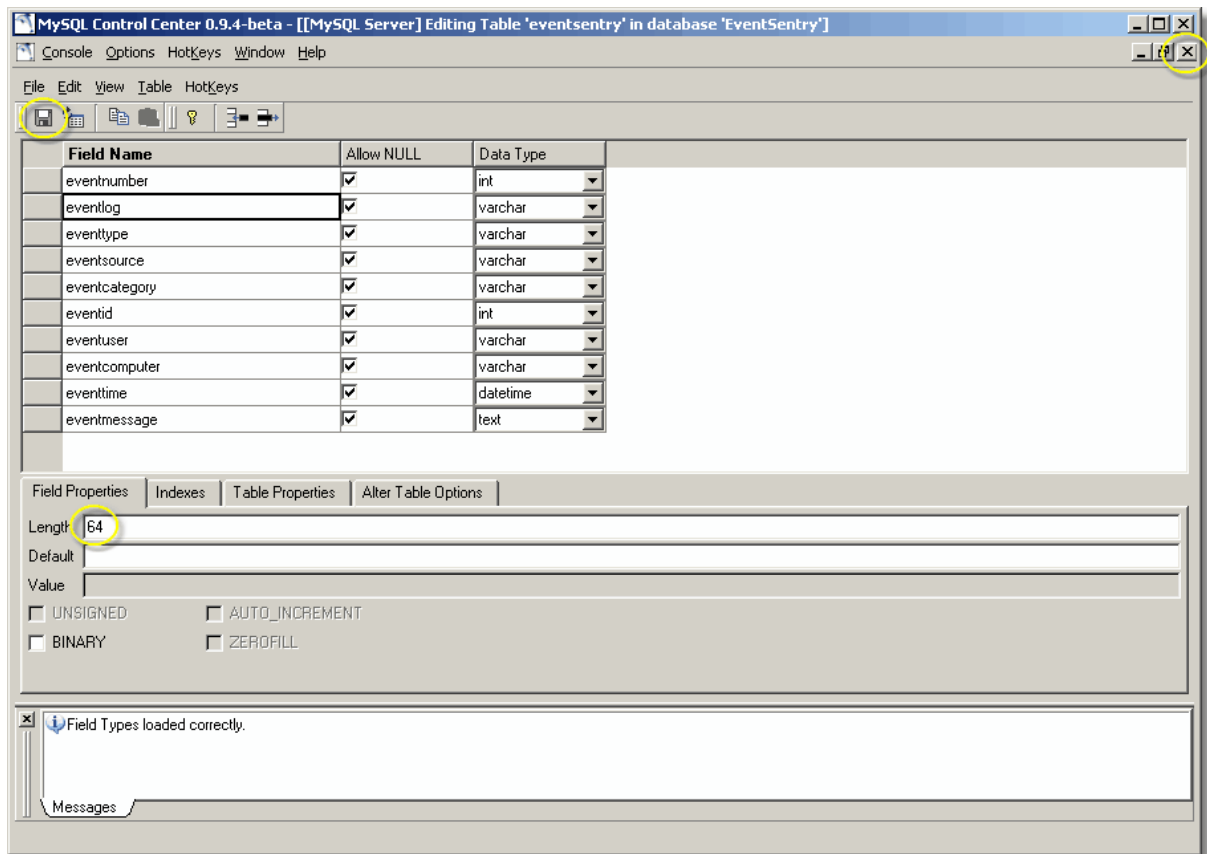


Creating the Table

To create the table we need to first connect to the database and then create the table. Right-click the database name and select **Connect**, the database icon should appear in green and you should see a **Tables** container once you are connected.



Right-click the **Tables** container and select **New Table**. In the next screen you can define the table fields, enter them as shown in the screenshot



When entering the **varchar** fields make sure that you specify the correct length, the table specification can be seen in the [SQL Shell](#) chapter. When you are finished click the **save** button in the upper left corner and enter the appropriate table name, **EventSentry**, when prompted. Now close the table window with the **X** in the upper right corner to return to the console manager.

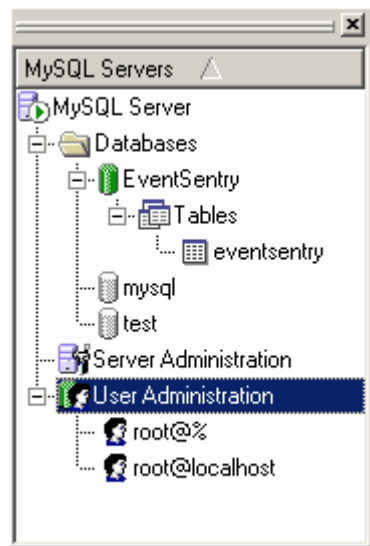
4.2.3 Creating Users

So far we have a new database and table, but the only user authorized to access this database is the user root, the only existing user! This needs to be changed and so we create two new user accounts:

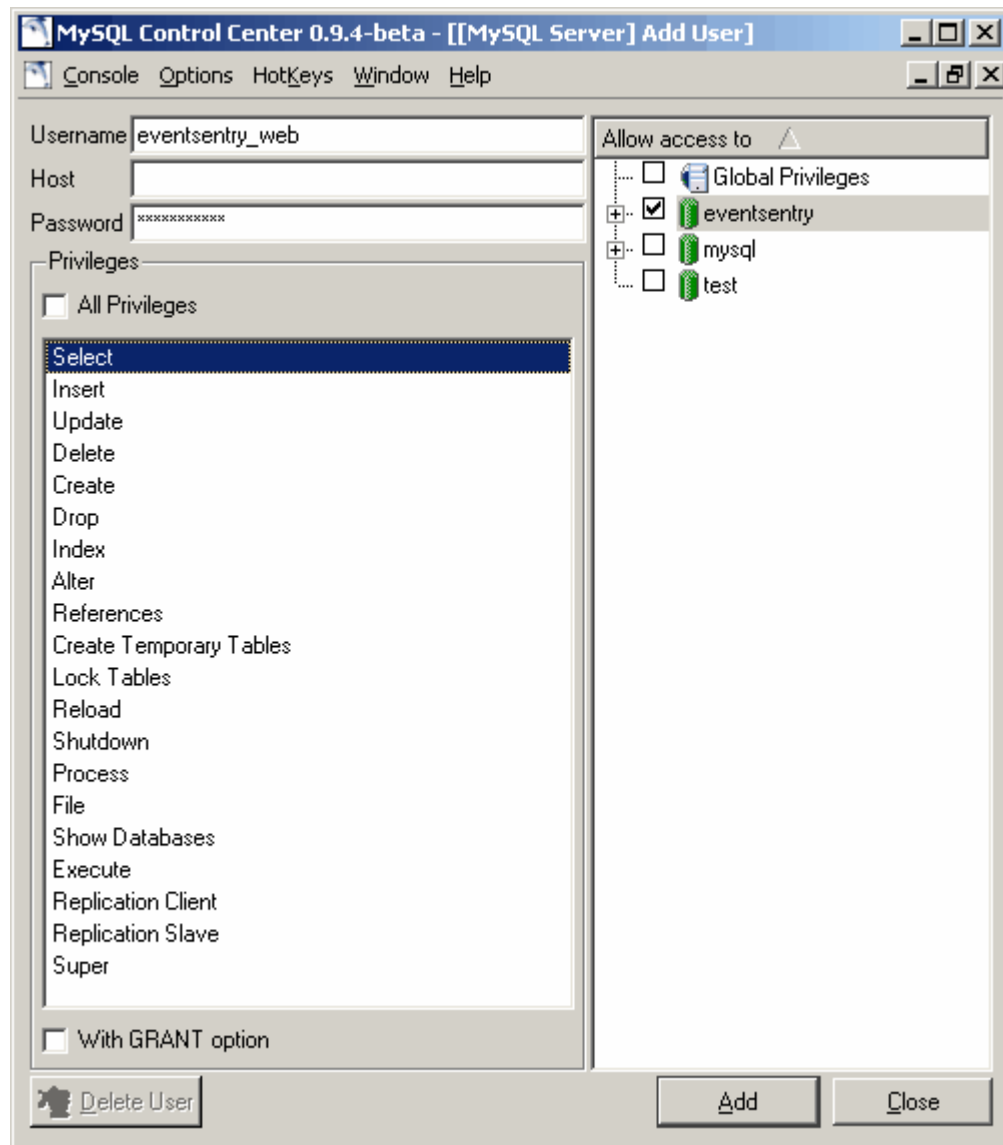
```
evententry_web
evententry_svc
```

The user **evententry_web** can be used by the ASP or PHP pages to connect and query the EventSentry database, while the **evententry_svc** user (svc = service) will be used by the various EventSentry services to connect and write event log data to the database. We are using two user accounts since both users will be used in different places (web server, servers) and performing different things. This way an intruder can't write to the database if he obtains the **evententry_web** users' password and vice versa.

In the console manager double-click the **User Administration** container to see all user accounts on the server. You should see different root accounts if you followed the instructions so far.



Right-click the **User Administration** container and select **New User** to add the user **eventsentry_web**. Then replicate the screenshot below where we give the user only **Select privileges** on the **EventSentry** database. Click **Add** and **Close**.



Now repeat this process for the **evententry_svc** user, only giving this user **Insert** privileges though, while leaving everything else the same. That's it, we created a database, a table and the users to access the database. You can see the whole thing in action below:

In this chapter we are assuming that you will be running a web server to query the EventSentry database from your web browser. You do not have to create the **event Sentry_web** user if will not be running a web server.

If you plan on running the web server on the same machine as the database server, then you can make your user account setup even more secure. Instead of typing the line

```
grant SELECT on EventSentry.* to event Sentry_web identified by  
'jUE42&@de';
```

type the line

```
grant SELECT on EventSentry.* to event Sentry_web@localhost identified by  
'jUE42&@de';
```

5 Appendix

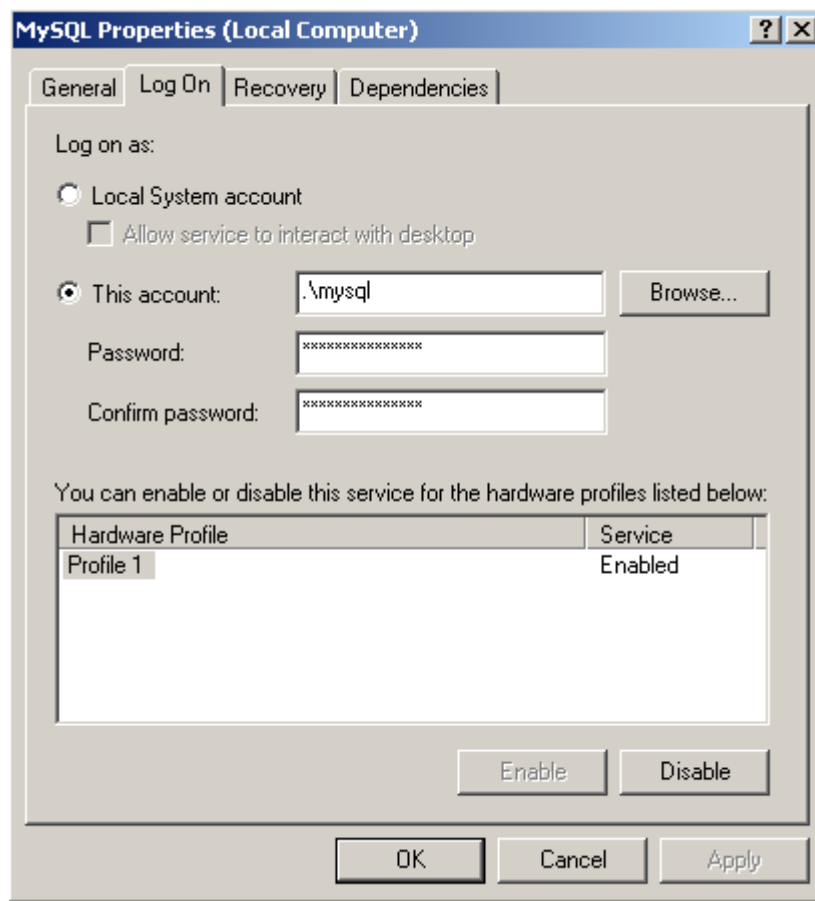
5.1 Security Tips

This chapter contains useful tips on how to secure a MySQL installation.

Changing the Service User Account

By default the MySQL service runs as the **LocalSystem** account which might a security problem if your database server is connected to non-trusted network. We recommend changing the service to run under a different user account.

1. Create a new user account, for example **MySQL** and give this user account **no special permissions**.
2. In the services control panel (Administrative Tools) double-click the MySQL service and select the **Log On** tab. Click on **This account** and select the user account you just created, then click OK.



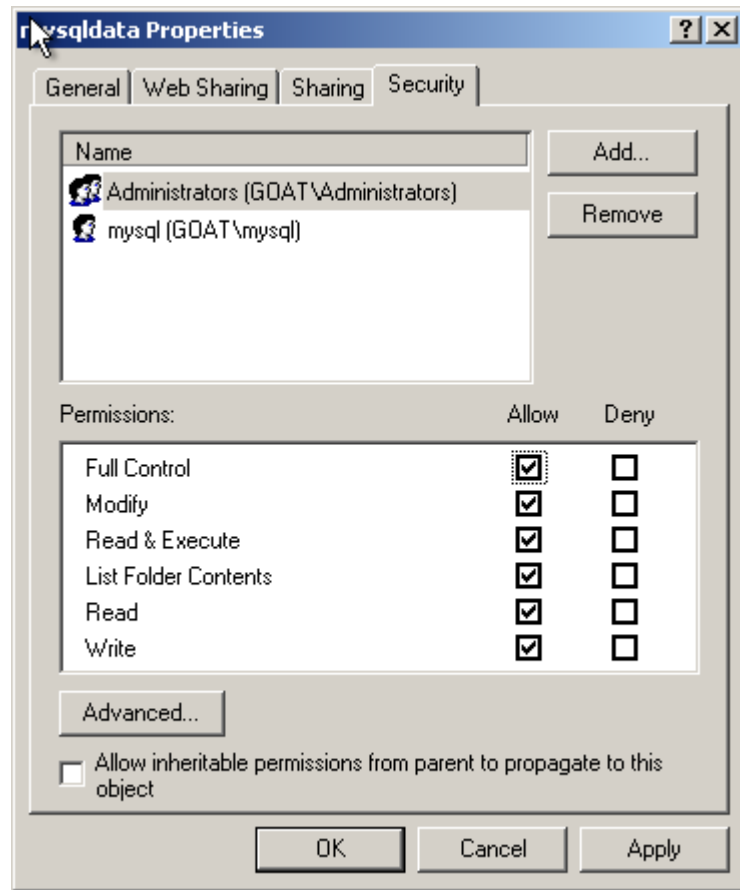
3. Restart the service to make the changes effective

Adjusting Access Rights on the Data File directory

We also need to change the access permissions on the directory where the data files are stored, `c:\mysql\data` by default. This directory gives everybody full permissions by default.

This directory should really only be accessible to Administrators and the MySQL user account we just

created. Configure the access rights as shown below, giving both **MySQL** and Administrators **FULL ACCESS** to the folder.



Changing the default TCP port 3306

It is sometimes a good idea to change the default port MySQL is listening on for remote connections. This can be done by adding the line

```
port = 9754
```

to the my.ini file in the [mysqld] section (mentioned in the [Data Storage](#) chapter of this file). You will need to restart the service for this change to become effective. You will also need to make sure that connections from untrusted networks, such as the Internet, are sufficiently protected with a firewall or similar device.

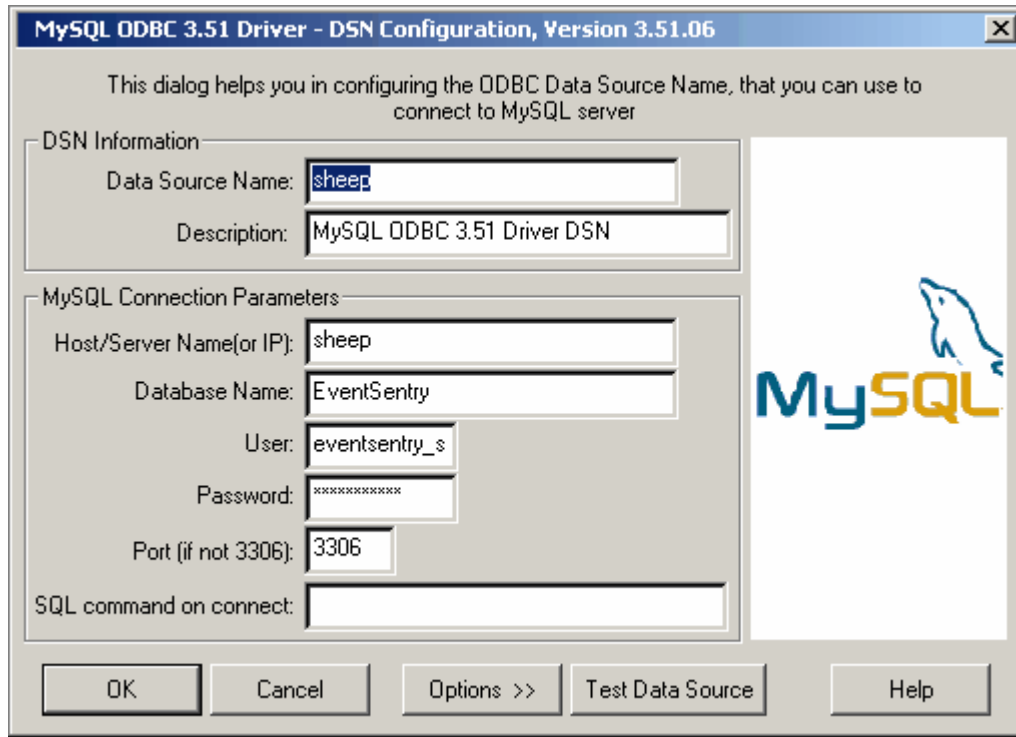
Staying Up To Date

You need to apply patches and software updates as soon as they become available to avoid security problems and worm infections. To stay informed subscribe to the **low volume** [MySQL Announcement mailing list](#).

5.2 ODBC Drivers

A common way to access databases for Windows clients (including the EventSentry agent) is to use ODBC. Naturally, MySQL ODBC drivers are not shipped with the Windows Operating System and must be installed manually on the clients.

You can download the latest ODBC drivers from <http://www.mysql.com/downloads/api-myodbc.html>. The installation and configuration are straightforward and do not require a reboot. The screenshot below shows the ODBC screen of MySQL.



We recommend using [AutoAdministrator](#) if you need to roll out MySQL ODBC drivers and DSN names to multiple machines.

5.3 Apache & PHP

When running the MySQL database then you might also want to run the Apache web server using PHP as a scripting interpreter. Our web pages for EventSentry are available for ASP and PHP.

Apache

Download the latest version of the 1.3 series of Apache from <http://httpd.apache.org/download.cgi>. Note that the 2.x series of Apache is preferable, however PHP does not officially support the 2.x release of Apache at the time of writing.

There are no special instructions for the installation of Apache, simply download and install the software. Make sure that IIS, if installed, is not running to avoid conflicts.

PHP

You can download the latest version of PHP from <http://www.php.net/downloads.php>. After installing the software you will need to make changes to both the Apache configuration file `httpd.conf` (usually found in the `c:\program files\apache group\apache\conf` directory) and the PHP configuration file `php.ini` (located in `%systemroot%`).

1. Add this line to the **httpd.conf** file below the other LoadModule lines:

```
LoadModule php4_module c:/php/sapi/php4apache.dll
```

2. Add this line to the **httpd.conf** file below the `ClearModuleList` command (this right below the line you just added)

```
AddModule mod_php4.c
```

3. Add these lines to the **httpd.conf** file, further down below where the `<IfModule>` sections start appearing

```
<IfModule mod_php4.c>  
    AddType application/x-httpd-php .php  
</IfModule>
```

4. When using the `eventsentry_*.php` files change the following line in the **php.ini** file

```
error_reporting = E_ALL; display all errors, warnings and notices
```

to the line

```
error_reporting = E_ERROR
```

to avoid non-critical warning messages being displayed.

5. Copy the files **php4apache.dll** and **php4ts.dll** from your php installation directory to the `%systemroot%\system32` directory. The former file can be found in the SAPI sub directory, the latter should be in the main installation directory.
6. Restart the apache service for these changes to become effective.