

A Visual Guide to Installing MySQL on Windows

A MySQL[®] White Paper

July 2009

Table of Contents

Introduction	3
Why MySQL on Microsoft Windows?	3
Installation.....	3
<i>Minimal Requirements.....</i>	<i>3</i>
<i>Step 1: Download</i>	<i>3</i>
<i>Step 2: Installation.....</i>	<i>5</i>
<i>Step 3: Configuration.....</i>	<i>7</i>
<i>Step 4: Testing the Server</i>	<i>11</i>
<i>Step 4: Adding a Sample Database.....</i>	<i>12</i>
<i>Step 5: Running Multiple MySQL Instances</i>	<i>13</i>
MySQL Connectors for Windows Developers.....	13
Easy to Manage and Monitor.....	14
<i>MySQL Enterprise Monitor</i>	<i>14</i>
<i>MySQL Query Analyzer.....</i>	<i>15</i>
MySQL on Windows Case Studies	16
<i>Adobe Relies on MySQL to Make Creative Professionals More Productive.....</i>	<i>16</i>
<i>NetQoS Delivers Distributed Network Management Solution with Embedded MySQL.....</i>	<i>16</i>
Conclusion	16
Resources.....	17
About MySQL.....	17

Introduction

The following guide is meant to make getting started with MySQL easily for Windows developers, system administrators and Microsoft SQL Server DBAs that are new to MySQL. This paper will guide you through the steps and tools needed to get up and running quickly with MySQL on Windows using familiar terms, comparisons, and language. Its goal is to put in place basic knowledge and building blocks you will need in conjunction with added references to resources as you continue to drill down further into the more advanced features of MySQL.

Why MySQL on Microsoft Windows?

The popularity of MySQL on Windows remains strong due to the fact that MySQL delivers:

- Lower TCO
- Ease of use
- Reliability
- Performance
- Fully featured database with no functional limitations

Another metric for judging the popularity of MySQL on Windows is to look at the average number of downloads for the MySQL server, tools and connectors for Windows from mysql.com. This number roughly averaged an astonishing 45,000 per day during the first six months of 2009. Finally, in Gartner's recent "Market Share: Relational Database Management System Software by Operating System, Worldwide, 2008" report they found that the growth for RDBMS running on Windows Server grew in 2007-2008 by 17.5%.

Installation

Minimal Requirements

MySQL supports all the current Windows versions from Microsoft with minimal additional requirements concerning software, hardware, storage or networking. Specifically, you can run MySQL on Windows XP, Windows Vista, Windows Server 2003 and Windows Server 2008. In general, you will need to install MySQL with Administrator privileges, but the MySQL process can run afterwards without any special administrative privileges. Many administrators also choose to run MySQL as a service so that it can be easily stopped and started using the Windows Service Control Manager. Finally, you will need TCP/IP support, as well as, sufficient local memory and storage to install and run your database.

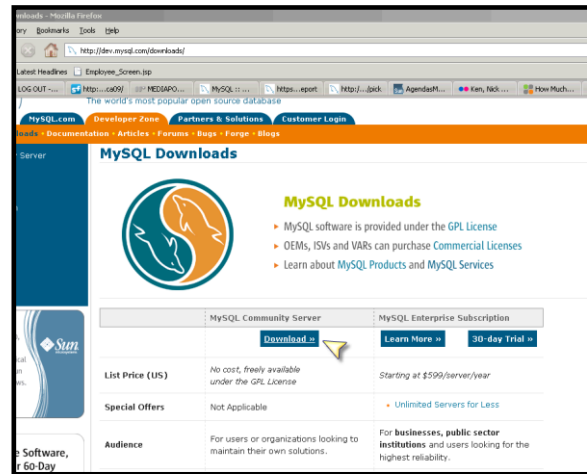
Step 1: Download

Getting started with MySQL on Windows is a very straight forward process. Unless you require an advanced or customized configuration of MySQL, the wizard driven Automated Installer will get you up and running on Windows in just a couple of minutes. The MySQL for Windows MSI packages can be downloaded from:

<http://dev.mysql.com/downloads/>.

You will see two MySQL Server installation options, MySQL Community and MySQL Enterprise. For the sake of this guide we will use the MySQL Community Server.

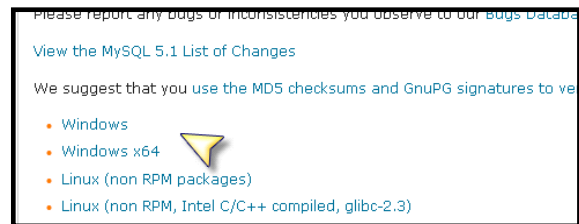
Click on **Download**



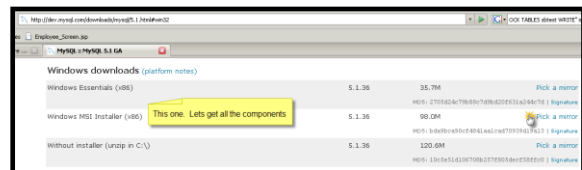
This takes you to takes you to:

<http://dev.mysql.com/downloads/mysql/5.1.html#downloads>

Select either **32** or **64 bit** depending on your system.



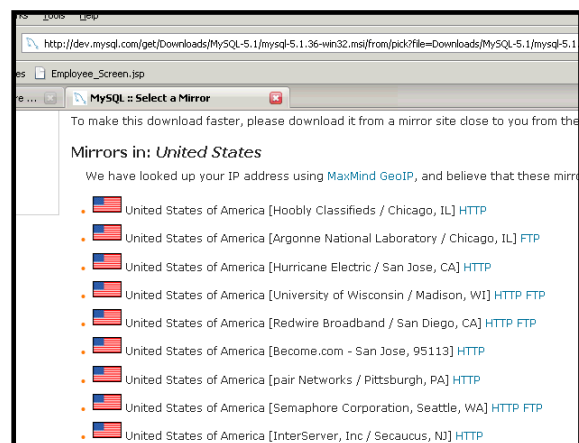
Next click on **Pick a mirror for Windows MSI Installer**



Note: The “Essentials” download simply excludes an Instance Manager tool, as well as Documentation, and Developer Components.

Click on **New User** and tell us about yourself.

Select a mirror that’s nearest to you geographically and save the file.



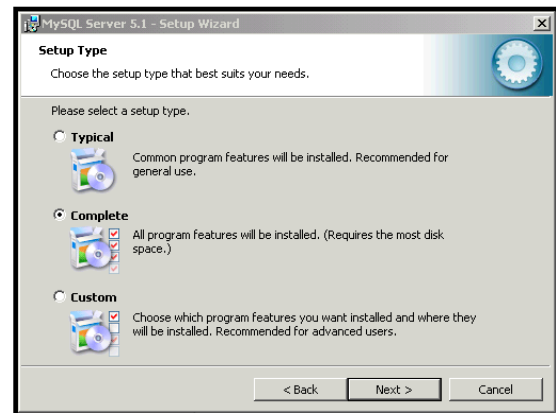
Once the MySQL Server Installer download is complete you are ready to **perform the installation**.

Step 2: Installation

Run the installer by double-clicking on the mysql-5.1.36-win32(2).msi file.



Next we **select a Typical, Complete or Custom** installation.

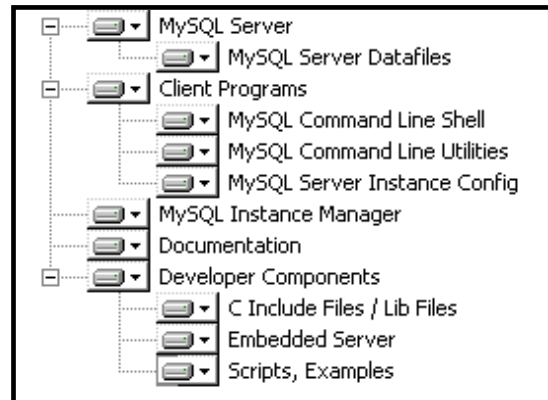


For your convenience, illustrated to the right is a list of the components contained in the installer.

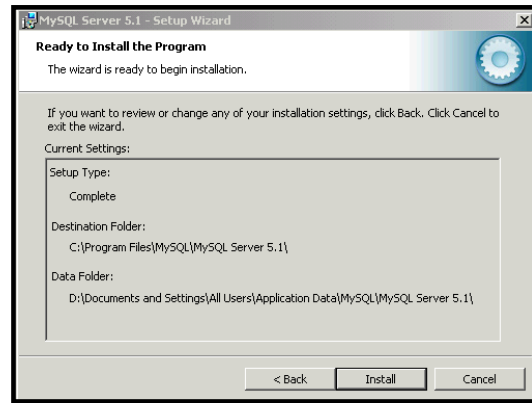
If you are a Developer **select Complete**, or if a DBA, then **select Typical**.

Click **Next**.

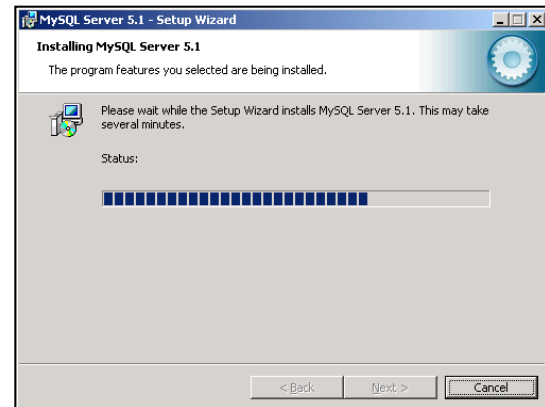
Note: The largest optional component (152 MB) that comes with the Complete installation that is often unneeded is the Embedded MySQL Server. The Embedded MySQL Server allows users to run MySQL embedded within their client application.



Verify the options you selected and **click install**.



Click next.



Click next.



Leave Configure the MySQL Server checked and then Finish.

The installation is complete and you are now ready to configure the server.

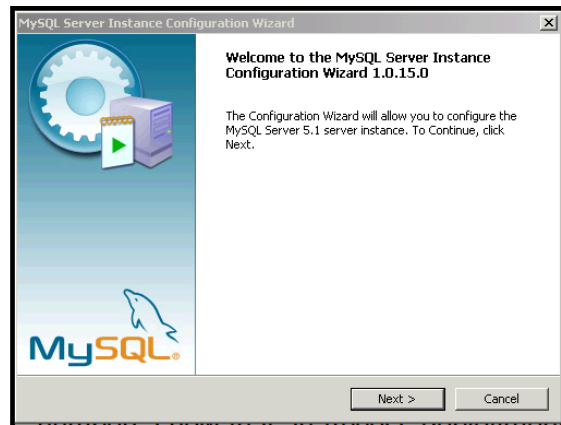


Step 3: Configuration

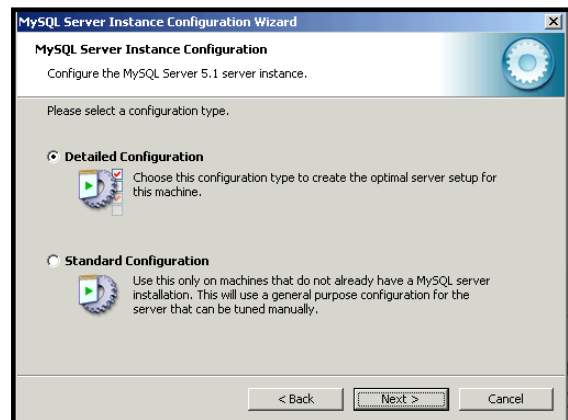
The initial configuration should now appear.

Note: If not then you can run it from the Windows Start Menu.

Start->MySQL->MySQL Server 5.1-> MySQL Server Instance Config Wizard



Click on next on the Welcome screen and **select the Detailed Configuration type**, then **click next**.



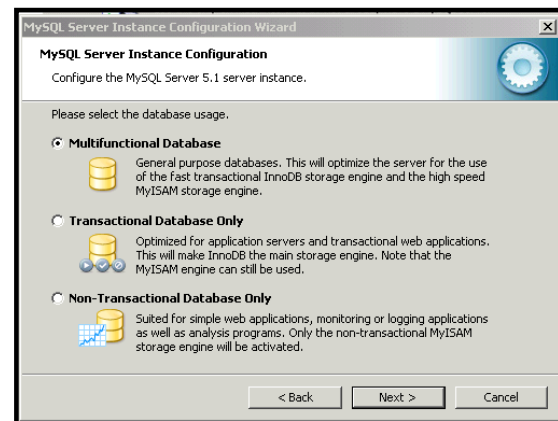
The next two screens are used to configure the MySQL Instance. The term *instance* refers to a running MySQL Server Process typically running as a Windows Service. The features of that MySQL Instance are determined by setting parameters (such as how much memory), startup options, and the active components for that MySQL service. Those parameters for the most part are set by the Instance Configuration Wizard within a file, *my.ini* in your installation directory (likely in C:\Program Files\MySQL\MySQL Server 5.1). If you want to make changes, you can rerun the Instance Configuration Wizard later.

For this trial run **select Developer** and **click next**.



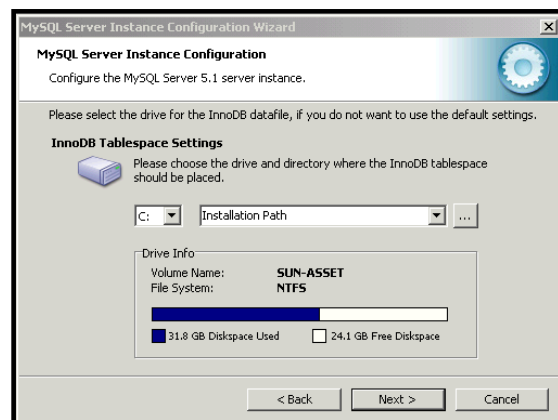
The options on this screen may likely present a new concept for some DBAs. Unlike other database products, MySQL supports many Database [Storage Engines](#), which can run at the same time under the same MySQL Instance.

Select Multifunction and **click Next** so you can experiment with tables using the MyISAM and InnoDB storage engines.

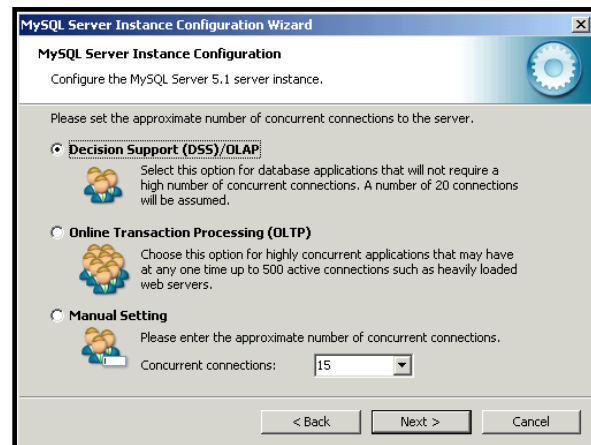


This step defines the location where your database data files will be stored for InnoDB. If you are on a system with high performance storage like a SAN or RAID device mounted, then **select the drive letter mapped to that drive**.

If you happen to have multiple disks you may also want to select a disk that has space available and where the OS is not installed.



Make your concurrent selection here based on the type of tests you'll run. For a simple first trial **select Decision Support (DSS)/OLAP**.



MySQL Server uses the network port 3306 by default. For now you probably should leave this as-is, but you can certainly change it later. If you do change it remember what you changed it to, as you will need to specify it when you connect this MySQL Server, and certainly don't change it to a port that's already in use.

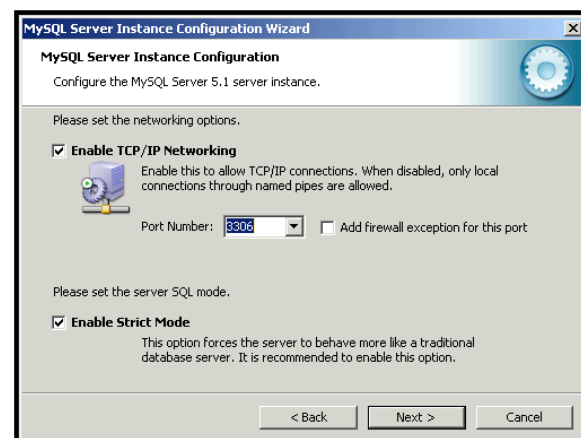
If you want to check your ports, from a Command prompt run:

```
netstat -an | find /i "listening"
```

You shouldn't see 3306 listed prior to the running the MySQL Server.

After MySQL is running, if you run netstat again you will see a line showing:

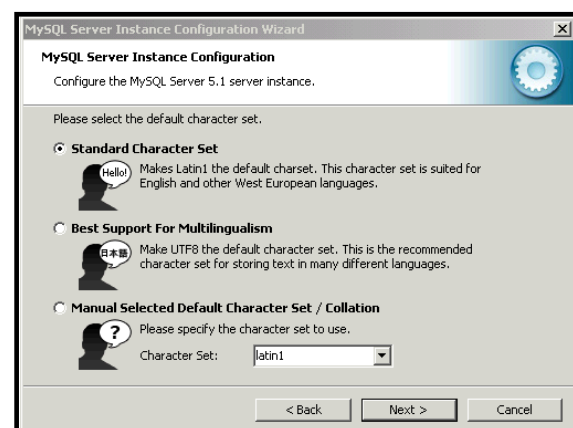
```
TCP 0.0.0.0:3306 0.0.0.0:0 LISTENING
```



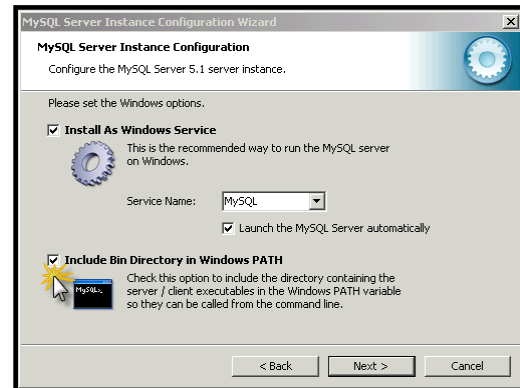
You may want to check the **Add firewall exception** for this port.

Click Next.

Then **select a Character Set** and **click Next.**



Select **Install As Windows Service** and include its command line tools in your **PATH**.

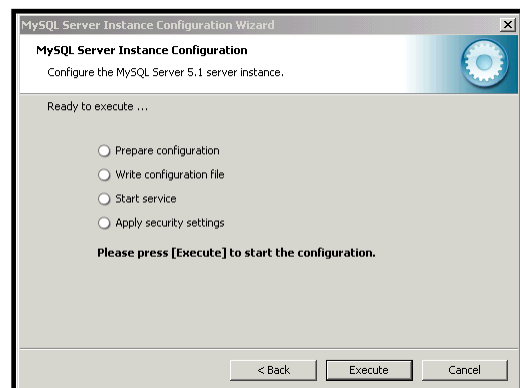


This next screen deserves some background as it can lead to confusion later, especially for Windows users. The MySQL Server administrator account is called “root”. It is equivalent to “sa” for those familiar with Microsoft SQL Server. If you are not familiar with SQL Server then it is similar to the local “Administrator” on a Windows Server. It is NOT an operating system account. **It is VERY important to remember the password you use here.** In this example we assume that you will perform admin tasks locally. Thus “Enable root access from remote machines” is not checked. It is more secure this way, but you can check this if you’d like to administer remotely. **Leave anonymous unchecked.**



Finally, **select Execute** and the MySQL Server will startup. That’s it.

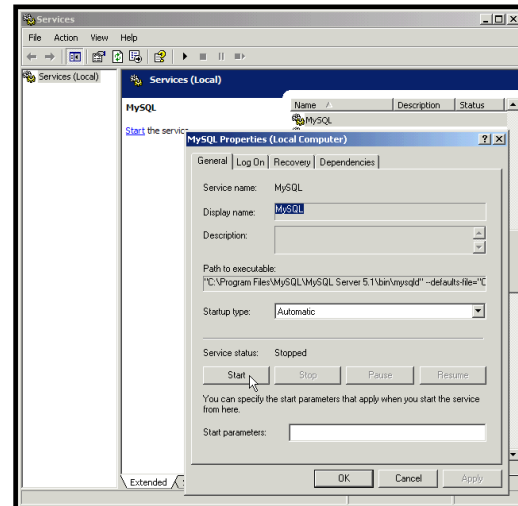
Next we will run a few simple checks.



Step 4: Testing the Server

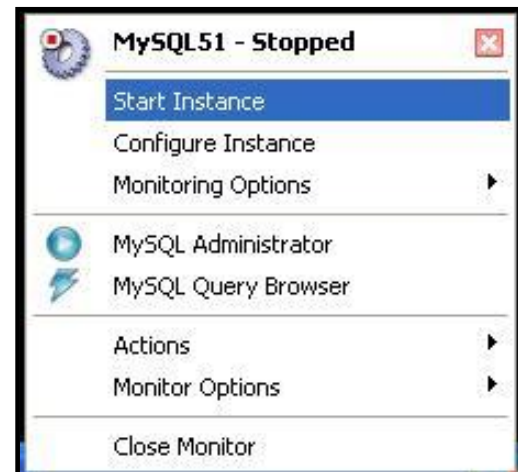
At this point in time your MySQL Server should be running as a service. If it's not then use the services dialog and to start it.

Note: If you didn't add it as a service then you can start it from the command line. This will vary depending on where you installed. But if you used the defaults the command will be something like this:



```
>"C:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld" --defaults-
file="C:\Program Files\MySQL\MySQL Server 5.1\my.ini"
```

MySQL on Windows does also come with its own Windows Systray application from which you can start, configure and perform other administrative tasks on your MySQL instance.



Now, lets just connect from the Windows Command line.

Type `mysql`. If it is not found that means you didn't add it to your path.

```
>set path
```

Will show if it is listed or not.

If it is not in your path, then change to the installed `dir` and the `bin` folder.

```
>cd "\\Program Files\MySQL\MySQL Server 5.1\bin"
```

Now **connect as root**. Using `-u` specifies the users, `-p` means intent to enter a password.

```
>mysql -u root -p
Enter password: <YOUR PASSWORD>
```

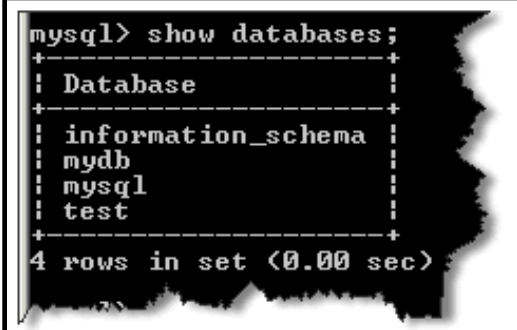
Next type **show databases**.

Be sure to end with a `;`, otherwise nothing will happen. The `;` indicates the end of the SQL statement. Also you can use `\g` like you would `GO` within Microsoft SQL Server.

```
mysql> show databases;
```

OR

```
mysql> show databases
mysql> \g
.
```



```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mydb        |
| mysql      |
| test       |
+-----+
4 rows in set (0.00 sec)
```

To get help with the mysql shell type:

```
mysql> help;
```

More specifics on a particular command, for example for more on show or create database. Etc.

```
mysql> help show;
mysql> help create;
mysql> help create database;
```

So if that worked then things are running properly so lets exit for now.

```
mysql> exit
```

Step 4: Adding a Sample Database

MySQL has a number of sample databases available for training and to serve as examples..

These databases can be downloaded from here - <http://dev.mysql.com/doc/#sampledb>

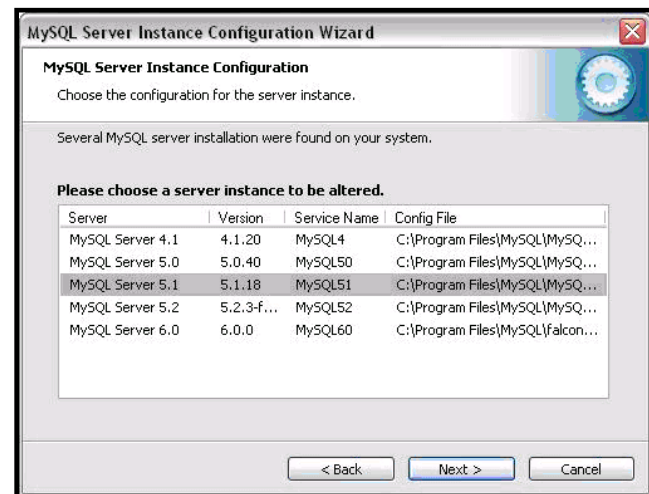
Let's start with Sakila. Click on Zip and save sakila-db.zip. Next unzip this file to C:\temp
It creates a directory named sakila-db

Follow the installation steps defined here

<http://dev.mysql.com/doc/sakila/en/sakila.html#sakila-installation>

Step 5: Running Multiple MySQL Instances

We should also point out that it is very easy to install and run multiple versions or instances of MySQL on a single box. To run multiple instances, you only need assign each instance a different port to run on. Managing different versions and instances is very easily done through either MySQL-supplied interfaces or the standard Windows service manager.



MySQL Connectors for Windows Developers

MySQL offers standard database driver connectivity for using MySQL with applications and tools that are compatible with industry standards ODBC and JDBC. This enables MySQL to work easily with standard development tools on Windows, Linux, Macintosh and Unix platforms. Any system that works with ODBC or JDBC can use MySQL. MySQL offers a variety of standardized database drivers regardless of the language choice for your application.

- **Connector/ODBC:** for Windows, Linux, Mac OS X, and Unix platforms.
- **Connector/J:** for Java platforms and development.
- **Connector/Net:** for .NET platforms and development.
- **Connector/MXJ:** MBean for embedding the MySQL server in Java applications.
- **Connector/C++:** for C++ development.
- **Connector/C (libmysql):** A client library for C development
- **MySQL native driver for PHP - mysqlnd:** The MySQL native driver for PHP is an additional, alternative way to connect from PHP 6 to the MySQL Server 4.1 or newer.

Of special interest for organizations developing applications using Microsoft's .Net, we offer **Connector/Net 6.0** which is the latest release of MySQL's fully managed ADO.Net provider to date.

This release contains some exciting features and improvements over previous versions:

- Significant speedups. In many areas, the connector now matches or surpasses the native C connector.
- Added UDF schema collection
- Better SQL and stored procedure tokenizing. Now all comment types are supported.
- Initial Entity Framework support

Also the Visual Studio integration has been completely overhauled yielding more features in a smaller package. Some of the new features include the following:

- Use of the traditional SQL Server buttons for keys, indexes, etc.

- Ability to generate a change script
- Use of Visual Studio's code editor which includes window splitting and syntax highlighting

Easy to Manage and Monitor

MySQL Enterprise Monitor

The MySQL Enterprise Monitor continuously monitors your MySQL servers (running on Windows or any other any other supported platform) and alerts you to potential problems before they impact your system. It's like having a "Virtual DBA Assistant" at your side to recommend best practices to eliminate security vulnerabilities, improve replication, optimize performance and more. As a result, the productivity of your developers, DBAs and System Administrators is improved significantly. For a free 30-day trial please visit:

<http://www.mysql.com/trials/>



MySQL Enterprise Monitor

MySQL Enterprise Monitor provides you with the following benefits:

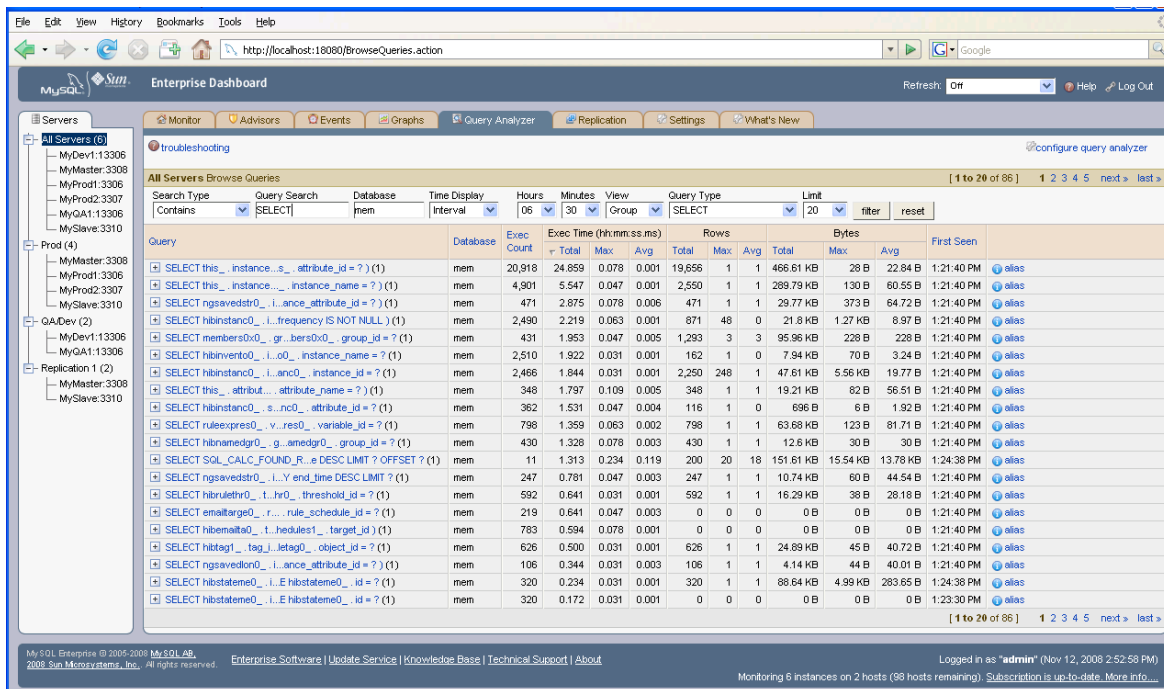
- Get a consolidated view into the health of all your MySQL servers

- Monitor over 600 MySQL and operating system variables with the Enterprise Dashboard
- Monitor MySQL sessions, connections, replication latency and more with 20+ graphs
- Improve application performance with the MySQL Query Analyzer
- Gain immediate visibility into your replication topologies through auto detection and grouping
- View real time master/slave performance using the MySQL Replication Monitor
- Customize the MySQL Enterprise Monitor for your specific needs using customization options
- Get notified of issues before they become costly outages using threshold driven alerts

The MySQL Enterprise Monitor is a distributed web application that is deployed within the safety of your firewall. It is comprised of a centralized Service Manager and lightweight Service Agent this is installed on each monitored MySQL server.

MySQL Query Analyzer

The MySQL Query Analyzer helps you improve application performance by monitoring query performance and accurately pinpointing SQL code that is causing a slow down. Queries are presented in an aggregated view across all MySQL servers so you can analyze your most expensive code by total executions, total run time, total data size and by using drill downs into sampled executions and EXPLAIN results. With the MySQL Query Analyzer, you can improve the SQL code during active development, and continuously monitor and tune the queries in production.



The screenshot shows the MySQL Enterprise Dashboard with the Query Analyzer tab selected. The main content area displays a table of queries with the following columns: Database, Exec Count, Exec Time (Total, Max, Avg), Rows (Total, Max, Avg), Bytes (Total, Max, Avg), and First Seen. The table lists various queries such as 'SELECT this_instance.s... attribute_id = ? (1)', 'SELECT this_instance... instance_name = ? (1)', and 'SELECT ngsavedstr0... ance_attribute_id = ? (1)'. The interface also includes a search bar, filters, and pagination controls.

MySQL Query Analyzer

MySQL Query Analyzer provides you with the following benefits:

- Continuously monitor queries running against all your MySQL Servers
- Quickly identify problem queries that impact the throughput of your applications
- View query details in drill down panels to analyze the root cause for performance degradation
- Fine-tune the problematic SQL code to regain peak performance

The MySQL Query Analyzer saves you time and effort in finding and fixing problem queries by providing:

- Aggregated view into query execution counts, run time, result sets across all MySQL servers with no dependence on MySQL logs or SHOW PROCESSLIST
- Sortable views by all monitored statistics
- Searchable and sortable queries by query type, content, server, database, date/time, interval range, and "when first seen"
- Historical and real-time analysis of all queries across all servers
- Drill downs into sampled query execution statistics, fully qualified with variable substitutions, and EXPLAIN results

MySQL on Windows Case Studies

Adobe Relies on MySQL to Make Creative Professionals More Productive

Adobe Systems is one of the largest software companies and is the leading provider of creative tools for print, web, interactive, mobile, video and film. Adobe embeds MySQL into several Adobe Creative Suite 3 components, including Adobe Acrobat CS3, Adobe® Bridge CS3, and Adobe® Version Cue® CS3 so that workgroups can work more efficiently on complex projects. For more information please visit:

http://www.mysql.com/why-mysql/case-studies/MySQL_CaseStudy_Adobe.pdf

NetQoS Delivers Distributed Network Management Solution with Embedded MySQL

NetQoS delivers products and services that enable some of the world's most demanding enterprises to improve network performance. American Express, Barclays, Boeing, Chevron, Cisco, Citrix, DuPont, Sara Lee, and Schlumberger are among the corporations that rely on NetQoS performance management solutions to ensure consistent delivery of business critical applications, monitor application service levels, troubleshoot problems quickly, contain infrastructure costs, and manage user expectations. To find the right embedded database solution to fit its innovative product architecture, NetQoS evaluated everything from flat-files to proprietary databases. NetQoS found that MySQL provided the ideal combination of performance, reliability, and ease of administration on Windows. For more information please visit:

<http://www.mysql.com/why-mysql/case-studies/mysql-netqos-casestudy.pdf>

For a complete list of case studies and other resources concerning organizations making use of MySQL on Windows, please visit:

<http://www.mysql.com/customers/operatingsystem/?id=109>

Conclusion

Windows remains a very popular MySQL development and production platform for ISVs, OEMs and organizations looking to lower the costs associated with developing and deploying databases. In this paper we visually covered step-by-step how to easily get a MySQL Server up and running on Windows. We also looked at how a subscription to MySQL Enterprise can assist in the ongoing management and monitoring of your MySQL Servers.

Resources

White Papers

<http://www.mysql.com/why-mysql/white-papers/>

Case Studies

<http://www.mysql.com/why-mysql/case-studies/>

Press Releases, News and Events

<http://www.mysql.com/news-and-events/>

Live Webinars

<http://www.mysql.com/news-and-events/web-seminars/>

Webinars on Demand

<http://www.mysql.com/news-and-events/on-demand-webinars/>

About MySQL

MySQL is the most popular open source database software in the world. Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems and packaged software -- including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube and Zappos.com. At <http://www.mysql.com>, Sun provides corporate users with commercial subscriptions and services, and actively supports the large MySQL open source developer community.

To discover how Sun's offerings can help you harness the power of next-generation Web capabilities, please visit <http://www.sun.com/web>.