MySQL Enterprise Monitor 2.0

MySQL Enterprise Monitor 2.0 Manual

Abstract

This manual documents MySQL Enterprise Monitor version 2.0.

Document generated on: 2010-11-03 (revision: 23545)

For more information on the terms of this license, for details on how the MySQL documentation is built and produced, or if you are interested in doing a translation, please visit MySQL Contact & Questions.

For additional licensing information, including licenses for libraries used by MySQL products, see Preface and Notes.

If you want help with using MySQL, please visit either the MySQL Forums or MySQL Mailing Lists where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the MySQL Documentation Library.

Table of Contents

List of Figures

List of Tables

Preface and Notes

This manual documents the MySQL Enterprise Monitor version 2.0. For license information, see the legal notice. This product may contain third-party code. For license information on third-party code, see Appendix A, *Licenses for Third-Party Components*.

Chapter 1. Introduction to MySQL Enterprise Monitor

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

The MySQL Enterprise Monitor serves as an automated assistant for MySQL database administrators. This service is designed to help administrators with their day-to-day tasks by monitoring MySQL servers and identifying potential problems. These features are designed to save the database administrator's time and effort by providing the information you need through a simplified interface.

An extension of existing MySQL Enterprise services, MySQL Enterprise Monitor monitors enterprise database environments and provides expert advice on how customers can tighten security and optimize the performance and uptime of their MySQL servers.

MySQL Enterprise Monitor helps administrators:

- Intelligently stay up to date with releases and bug fixes.
- Know what's going on with their system.
- Manage day-to-day database maintenance tasks.
- Improve the performance of their system.
- Manage and prevent crises.

The MySQL Enterprise Monitor was designed to tackle the job of managing the performance of any number of MySQL database servers, regardless of their physical or geographical location. Although MySQL Enterprise Monitor can easily track just a handful of MySQL servers, the service is specifically designed to greatly curtail the time it takes to get a handle on the availability and performance levels of many database servers at once.

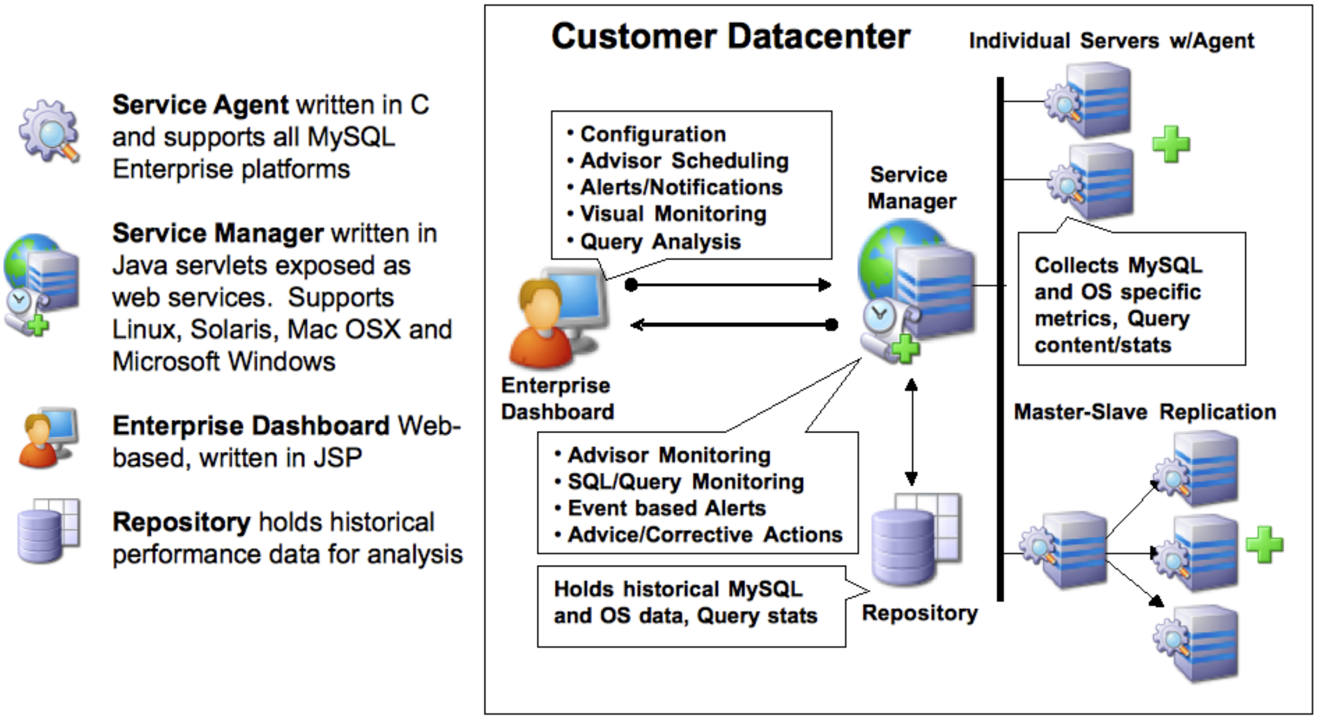
The MySQL Enterprise Monitor does this by providing an web-based interface—called the Enterprise Dashboard—that serves as the portal for viewing information about your MySQL database servers. MySQL professionals can manage all their servers by group or individually if need be.

The Enterprise Dashboard web interface does not have to be installed on individual desktops, but is instead available from a centrally located machine that serves as the main location for the Monitoring and Advisory service.

1.1. Overview of the Service

The MySQL Enterprise Monitor is a collection of components that work together to monitor and help administer your MySQL server installations. This service includes server management agents, advisors, and a central MySQL Enterprise Service Manager, all working in tandem with the MySQL Enterprise to keep your MySQL servers secure and up to date. All of this is controlled through the MySQL Enterprise Dashboard—a lightweight web-based interface that gives you complete control of your MySQL servers from any location.

Figure 1.1. MySQL Enterprise Monitor Architecture


Service Agent written in C and supports all MySQL Enterprise platforms
Service Manager written in Java servlets exposed as web services. Supports Linux, Solaris, Mac OSX and Microsoft Windows
Enterprise Dashboard Web-based, written in JSP
Repository holds historical performance data for analysis
Customer Datacenter
Individual Servers w/Agent
• Configuration
• Advisor Scheduling
• Alerts/Notifications
• Visual Monitoring
• Query Analysis
Service Manager
Collects MySQL and OS specific metrics, Query content/stats
Enterprise Dashboard
Master-Slave Replication
• Advisor Monitoring
• SQL/Query Monitoring
• Event based Alerts
• Advice/Corrective Actions
Holds historical MySQL and OS data, Query stats
Repository

The service is made up of a number of components, including the Monitor Agent (MySQL Enterprise Monitor Agent), the Service Manager (MySQL Enterprise Service Manager), the Enterprise Dashboard, the Repository and the Advisors.

1.1.1. The Service Architecture

The MySQL Enterprise Monitor is powered by a distributed web-based application that is deployed within the confines of the corporate firewall. The Enterprise Dashboard provides the interface to the server data, advisor notifications, live information and communication with the MySQL Enterprise Update Service.

Subscribers are kept up to date about the latest releases of the MySQL server or issues that may affect their specific implementation of MySQL by using the MySQL Enterprise Update Service. This same mechanism is used to notify MySQL Enterprise Monitor users of updates to the application or to the MySQL Advisors and Rules. The various components are described below.

- **Monitor Agent**

  Monitor Agents are the foot soldiers of the MySQL Enterprise Monitor; they monitor each MySQL server. Running as a Windows service or Unix daemon, the Agent uses a combination of MySQL specific commands, SQL queries, and custom scripts to collect and report MySQL server or operating system (OS) specific data. The Monitor Agent initiates a “heartbeat” to the Service Manager on a regular basis to ensure specific MySQL server and OS level data collections are current.

  In the overall architecture, the Monitor Agent is the only component of the MySQL Enterprise Monitor that establishes or maintains a connection with the monitored MySQL Server. As with any MySQL client, the Monitor Agent is authenticated on the monitored MySQL server and requires a user name and password to establish a connection.

  In addition, the MySQL Enterprise Monitor Agent also provides a proxy service that allows for information about queries to be captured and reported as part of the Query Analyzer functionality. The MySQL Enterprise Monitor Agent accepts client connections and forwards the SQL statements on to the server and returns the results. In the background, the agent is collecting information about the query execution, row counts, times and other data so that queries and their execution can be monitored.

- **Service Manager**

  The Service Manager is the heart and soul of the MySQL Enterprise Monitor. It is built on a collection of Java services hosted on a single Windows or Unix server. The Service Manager interacts with all of the Monitor Agents under its domain to collect MySQL server and OS level data for each of the monitored MySQL servers.

  The Service Manager performs many duties including:

  - Enterprise Dashboard, the main interface to the MySQL Enterprise Service Manager.
  - Autodiscovery of monitored MySQL Servers.
  - Creation and management of Monitor Agent tasks.
  - Storage of data collections from Monitor Agents.
  - Monitoring of key MySQL server and OS level metric data collections.
  - Reporting MySQL best practice events and violations.
  - Providing MySQL expert advice for MySQL best practice violations.
  - Autodiscovery of replication topology (Not available for all subscription levels)

- **The Repository**

  The Repository is built on MySQL 5.0.*x* and is used to store MySQL server and OS level data collections for each of the monitored MySQL Servers. This information is used by the Service Manager to evaluate and report the health and status of the monitored MySQL environment(s).

- **The Enterprise Dashboard**

  The MySQL Enterprise web client provides the graphical user interface (GUI) for the MySQL Enterprise Monitor. This interface is the primary means of monitoring the state of your MySQL servers, identifying rule violations and providing advice on how best to address and correct any underlying issues.

  This interface also provides an easy means of configuring advisors, adding users, creating notification groups, and receiving updates from MySQL Enterprise.

1.1.2. Service Features

The key features of the MySQL Enterprise Monitor can be summarized as follows:

- Group-level or Server-level management options
- Enterprise Dashboard for managing all MySQL Servers from a consolidated console
- Monitoring page for “at a glance” global health check of key systems
- MySQL-provided Advisors and Advisor Rules for enforcing MySQL Best Practices
- Advisor Rule Scheduling for unattended operations
- Customizable Thresholds and Alerts for identifying Advisor Rule violations
- User-Defined Advisor Rules
- Event/Alert History browser for researching advisor-specific events and annotations
- Query Analyzer functionality allowing you to monitor the execution times, row counts and other data about queries executed on your MySQL server.

These features are presented through the MySQL Enterprise Dashboard which is made up of six main pages:

- The **MONITOR** page comprises:
  - The **SERVER TREE**: Easily navigate monitored servers

    - The **Graphing**: This capability is built in so you can quickly assess critical functions such as activity, performance metrics, and number of connections

    - The **Heat Chart**: Color-coded buttons provide key operating system and database metrics

- The **Advisors** page

  This page shows the advisors that are currently scheduled. There are advisors for a variety of topics such as security and indexing. Users can add, edit, or create their own advisors.

- The **Events** page

  This page shows rule violations, indicating the server, severity, and time of occurrence. A number of filter options are available, allowing various views of events.

- The **Graphs** page

  Use this page to view all the available graphs and to adjust the scale of the graphs, for a more or less detailed view as the situation requires.

- The **Query Analyzer** page

- The **Replication** page

  Use this page to keep track of your masters and their slaves (Not available for all subscription levels)

- The **Settings** page

  On this page you configure servers, users, email addresses, and notification groups. Entering a user name and password for MySQL Enterprise provides automatic updates.

1.1.3. Security

Using the Tomcat/Apache web server for the user interface allows an administrator to configure the web server to meet any security regulations. The MySQL Enterprise Monitor architecture is designed to be as secure as possible, even when monitoring systems outside of the local network.

Communications between the MySQL Enterprise Monitor Agent and MySQL Enterprise Service Manager can be protected by Secure Socket Layer (SSL) encryption and server and agent can use SSL certificates to provide authentication and prevent spoofing.

The MySQL Enterprise Monitor Agent is like a web browser — it is an HTTP client application that initiates all communication with the MySQL Enterprise Service Manager. If the server requires action from the agent, it must wait until the agent next initiates contact and sends its request in a response. This means you do not need to open an inbound port on the machine on which the agent is running because it does not listen for requests. However, an outbound port must be open for the agent to contact the MySQL Enterprise Service Manager.

As an additional security feature, each Agent can have a separate Advisory Service login which minimizes exposure should any one agent be compromised.

1.2. Conventions Used in This Document

This document uses certain typographical conventions:

- `Text in this style` is used for SQL statements; database, table, and column names; program listings and source code; and environment variables. Example: “To reload the grant tables, use the `FLUSH PRIVILEGES` statement.”

- **`Text in this style`** indicates input that you type in examples.

- `Text in this style` indicates the names of executable programs and scripts, examples being `mysql` (the MySQL command line client program) and `mysqld` (the MySQL server executable).

- *`Text in this style`* is used for variable input for which you should substitute a value of your own choosing.

- File names and directory names are written like this: “The global my.cnf file is located in the /etc directory.”
- Character sequences are written like this: “To specify a wildcard, use the ‘%’ character.”
- *Text in this style* is used for emphasis.
- **Text in this style** is used in table headings and to convey especially strong emphasis.

When commands are shown that are meant to be executed from within a particular program, the prompt shown preceding the command indicates which command to use. For example, shell> indicates a command that you execute from your login shell or from the command line in Windows:

```
shell> type a shell command here
```

The “shell” is your command interpreter. On Unix, this is typically a program such as sh, csh, or bash. On Windows, the equivalent program is command.com or cmd.exe, typically run in a console window.

When you enter a command or statement shown in an example, do not type the prompt shown in the example.

Sometimes, what appears on one line in a console window cannot be represented in the documentation on a single line. In cases such as this the character ‘»’ is used. For example:

```
Please specify the directory where the MySQL Enterprise Monitor »
 will be installed.
```

Where Unix commands are concerned, the continuation character ‘\’ is used. Doing this allows commands to be copied and pasted to the command line verbatim. For example:

```
shell> /opt/mysql/enterprise/agent/bin/mysql-monitor-agent -f \
 /opt/mysql/enterprise/agent/etc/mysql-monitor-agent.ini
```

SQL keywords are not case sensitive and may be written in either case. This document uses uppercase.

In syntax descriptions, square brackets (‘[’ and ‘]’) indicate optional words or clauses. For example, in the following statement, IF EXISTS is optional:

```
DROP TABLE [IF EXISTS] tbl_name
```

When a syntax element consists of a number of alternatives, the alternatives are separated by vertical bars (‘|’). When one member from a set of choices *may* be chosen, the alternatives are listed within square brackets (‘[’ and ‘]’):

```
TRIM([[BOTH | LEADING | TRAILING] [remstr] FROM] str)
```

When one member from a set of choices *must* be chosen, the alternatives are listed within braces (‘{’ and ‘}’):

```
{DESCRIBE | DESC} tbl_name [col_name | wild]
```

An ellipsis (...) indicates the omission of a section of a statement, typically to provide a shorter version of more complex syntax. For example, INSERT ... SELECT is shorthand for the form of INSERT statement that is followed by a SELECT statement.

An ellipsis can also indicate that the preceding syntax element of a statement may be repeated. In the following example, multiple *reset_option* values may be given, with each of those after the first preceded by commas:

```
RESET reset_option [,reset_option] ...
```

Commands for setting shell variables are shown using Bourne shell syntax. For example, the sequence to set the CC environment variable and run the configure command looks like this in Bourne shell syntax:

```
shell> CC=gcc ./configure
```

If you are using csh or tcsh, you must issue commands somewhat differently:

```
shell> setenv CC gcc
shell> ./configure
```

Throughout this document the term ‘Unix’ is used to describe any Unix or Unix-like operating system. For an up-to-date list of operating systems supported by the MySQL Enterprise Monitor please see the http://www.mysql.com/products/enterprise/.

Chapter 2. Installation and Upgrades

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

This chapter describes the process of installing the MySQL Enterprise Monitor on all operating systems. A working installation requires the installation of a MySQL Enterprise Service Manager, the MySQL Enterprise Advisors and one or more MySQL Enterprise Monitor Agents. Simply described, the agent inspects the MySQL server it is monitoring, reports to the Service Manager, and the results are interpreted by the advisors and displayed in the MySQL Enterprise Dashboard for viewing in a web browser.

One Monitor Agent is installed for each MySQL server that is being monitored. The Monitor Agent usually runs on the same machine that hosts the monitored MySQL server but it may run on any machine that has access to both the monitored MySQL server and the MySQL Enterprise Dashboard. The agent reports its findings to the Service Manager and these results are interpreted by Advisors and displayed in the dashboard. The end user opens a web browser to view the information presented in the dashboard. The Service Manager and dashboard run on the same machine and both have access to a local MySQL server installed as part of the MySQL Enterprise Monitor. This server is known as the repository and provides storage for the data provided by the agent.

Installation is a three step process:

1. Install and start the Service Manager on the monitoring system. See Section 2.2, “Service Manager Installation”.

2. Configure the Service Manager, see Section 2.3, “Monitor Agent Installation”.

3. Start the MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent instances, and then use the MySQL Enterprise Dashboard to install the Advisors and complete the configuration and installation. See Section 2.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

Depending on your configuration and environment, you will need to download a number of different components and files from MySQL Enterprise website available on the download page. These include:

- MySQL Enterprise Service Manager and MySQL Enterprise Dashboard for the platform that you intend to execute the MySQL Enterprise Service Manager on. These are named `mysqlmonitor-2.1.0.1096-linux-x86_64-installer.bin`, with the appropriate version and platform name. If you are performing an upgrade, download the upgrade installer, named `mysqlmonitor-2.1.0.1096-linux-x86_64-update-installer.bin`.

- One or more MySQL Enterprise Monitor Agent, one for each MySQL Server that you want to monitor. You should download an installer package for the right platform for the MySQL server you want to manage. Agent installers are available with the name `mysqlmonitoragent-2.1.0.1093-linux-debian3.1-powerpc-installer.bin`. Upgrade installers to update an existing MySQL Enterprise Monitor Agent installation are named `mysqlmonitoragent-2.1.0.1093-linux-debian3.1-powerpc-update-installer.bin`.

- *Optional*

  A product key file for MySQL Enterprise Service Manager. If your MySQL Enterprise Service Manager has internet connectivity, your product key can be downloaded automatically during the initial phase of configuration directly from the MySQL Enterprise website. For more information, see Section 2.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

- *Optional*

  An advisor bundle equal to the level of your MySQL Enterprise subscriptiion (Platinum, Gold, or Silver), which is required for MySQL Enterprise Service Manager. If your MySQL Enterprise Service Manager has internet connectivity, your advisor bundle can be downloaded automatically during the initial phase of configuration directly from the MySQL Enterprise website. For more information, see Section 2.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

For information on the installation requirements for different platforms, see Section F.3, “Installation Requirements”.

2.1. User Roles

Prior to installation you will need to have at hand credentials for access to the MySQL server you plan to monitor and also your MySQL Enterprise credentials. *During* installation and when first logging in, you will set up a variety of users with different roles and credentials. This can become confusing. This section outlines the various users associated with the MySQL Enterprise Monitor and gives a brief description of their roles.

2.1.1. Existing Users

The **MySQL Enterprise user**: These are the credentials you use to log in to the MySQL Enterprise web site. You will need them to acquire the Advisor files and receive updates and, if necessary, acquire a product key.

The **MySQL user**: For Monitor Agents to report the status of a MySQL server they must have privileges on that server. To perform all functions an agent must have `SHOW DATABASES`, `REPLICATION CLIENT`, `SUPER`, `CREATE`, and `SELECT` privileges. In short, the Monitor Agent needs to have read access to all data. Details about this account are given in Section 2.3.1, "Creating a MySQL User Account for the Monitor Agent".

2.1.2. User Created During Installation

The **Repository user**: This user is the only user in the `user` table in the `mysql` database in the bundled MySQL server. To avoid confusion with monitored MySQL servers, this server is referred to throughout this document as the `repository`. The repository user can log in from `localhost` using the password specified during installation and has all privileges on all databases. These credentials are used to create the repository and its tables and to record data in them. During installation the default value for the user name for this role is `service_manager`. No default password is specified. You can use these credentials to manage the repository from the command line or when using a program such as MySQL Administrator.

During installation the file `configuration_report.txt` is created. Reference this file for the credentials of the repository manager. After the MySQL Enterprise Service Manager is installed, look for this file in the following directories:

- Windows: `C:\Program Files\MySQL\Enterprise\Monitor`
- Unix: `/opt/mysql/enterprise/monitor`
- Mac OS X: `/Applications/mysql/enterprise/monitor`

2.1.3. Users Created on First Log-in

The **Root user**: This user is the administrator of the dashboard. The first time you log in to the dashboard you must log in as this user. The default user name for this user is `admin`. There is no default password for this user.

The **Agent user**: The Monitor Agent needs to report the status of the MySQL server it is monitoring. For this reason it needs to log in to the dashboard. The default user name for this user is `agent`. There is no default password for this user.

> **Note**
>
> The Monitor Agent has two roles in the MySQL Enterprise Monitor; it must have access to the dashboard and to the MySQL server it is monitoring. For a description of the agent as a MySQL user see Section 2.1.1, "Existing Users".

2.2. Service Manager Installation

The MySQL Enterprise Service Manager is the core element of the MySQL Enterprise Monitor. The installation process for this element is completely self-contained, but the installation includes the following components:

- Apache Tomcat
- MySQL Server
- Java VM

> **Note**

> After installation you can determine the version number of the various components by entering `http://server_name:18080/main?command=list_versions` into the web browsers address bar.

During installation, versions of MySQL and Tomcat will be installed onto the machine. The installer automatically provides default network ports that are different from standard installation for these applications. You can change the ports during installation.

During installation, default values are shown for user names and ports. This is for your convenience only; you may choose different values. The installer detect ports that are already in use and allows you to select different ports.

> **Warning**
>
> The MySQL Enterprise Service Manager version 2.0 requires agents using 2.0 or higher.

All the installations share the same basic configuration parameters that you will be asked to confirm during installation. Before you start your installation, please review the section on these common paramaters, then proceed to section specific to your installation platform. For details of the common parameters, see Section 2.2.1, “Service Manager Installation Common Parameters”. For information on installation under Windows, see Section 2.2.2, “Service Manager Installation on Windows”, for Mac OS X see Section 2.2.3, “Service Manager Installation on Mac OS X”, and for Unix/Linux, see Section 2.2.4, “Service Manager Installation on Unix”.

2.2.1. Service Manager Installation Common Parameters

All installations of the Service Manager install the Tomcat and MySQL applications using the same basic set of parameters. The defaults provided by the installation process are designed to be unique so that they do not interfere with existing installations of either product. However, you should check these parameters before installation to ensure that you do not experience any problems.

The common parameters are divided into those applying to the Tomcat server, and the MySQL server (Repository Configuration):

- **Tomcat Server Options**

  - Tomcat Server port: The default port that the Tomcat server will use when listening for connections. If you change this option, then the port that you need to use when connecting to the Service Manager must be modified accordingly. The default value is 18080.

    > **Note**
    >
    > If you are not currently running a web server on port 80 you may find it more convenient to use the well known port rather than `18080`. Since port `80` is the default for a web server, you can then open the dashboard without specifying a port.

  - Tomcat Shutdown port: The port used by the management scripts that is used to shut the Tomcat server down when you need to stop the Service Manager. The default value is 18005.

  - Tomcat SSL Port: The standard port used to connect to the Service Manager when you want to use Secure Sockets Layer (SSL) encrypted communication. The default value is 18443.

- **Repository Configuration (MySQL Server)**

  - Repository Username: The user name created and used to store information within the MySQL server to hold the information used by the Service Manager. In normal use, you should not need to use or modify this information, but it may be required if you have a support issue. The default value is `service_manager`.

  - Repository User password: The password to be used for the Repository Username. This should be set to a secure password so that the repository data is secure.

The information that you configure during installation will always be recorded within the `configuration_report.txt` file within the installation directory for the Service Manager.

> **Caution**
>
> Because the information stored within the `configuration_report.txt` file is in plain text, the Repository user name and password information are also exposed within this file. Make sure that the installation directory and file are secure that they can only be accessed by those users who would need to use the information.

2.2.2. Service Manager Installation on Windows

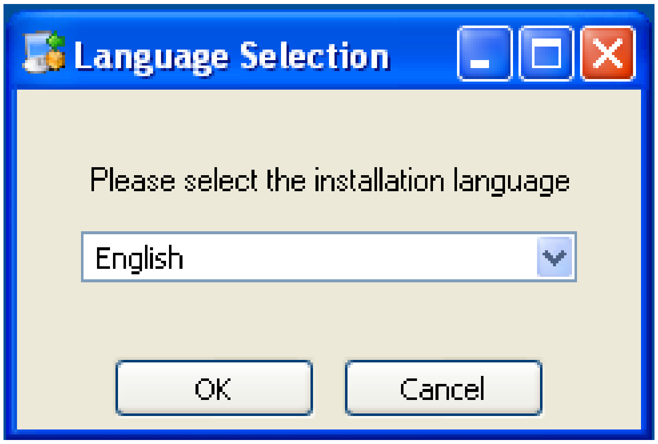
On Windows the installation modes are `win32` and `unattended` only. `unattended` mode is especially useful if you are doing multiple installations. For more information on this topic see Section 2.4, “Unattended Installation”.

> **Note**
>
> To install the Service Manager as a Windows service, you must do the installation as a privileged user.
>
> On Windows Vista or later, if user account control is on, an operating system dialog box requests confirmation of the installation.

To install the Service Manager on Windows, find the executable file named `mysqlmonitor-`*`version`*`-windows-installer.exe` (where *`version`* represents the three-part version number).

1. Double-click the MySQL Monitor installer. You should be presented with the Language Selection prompt. Select the language to use for the installer and then click OK.

Figure 2.1. MySQL Enterprise Monitor: Installing Monitor on Windows: Language Selection



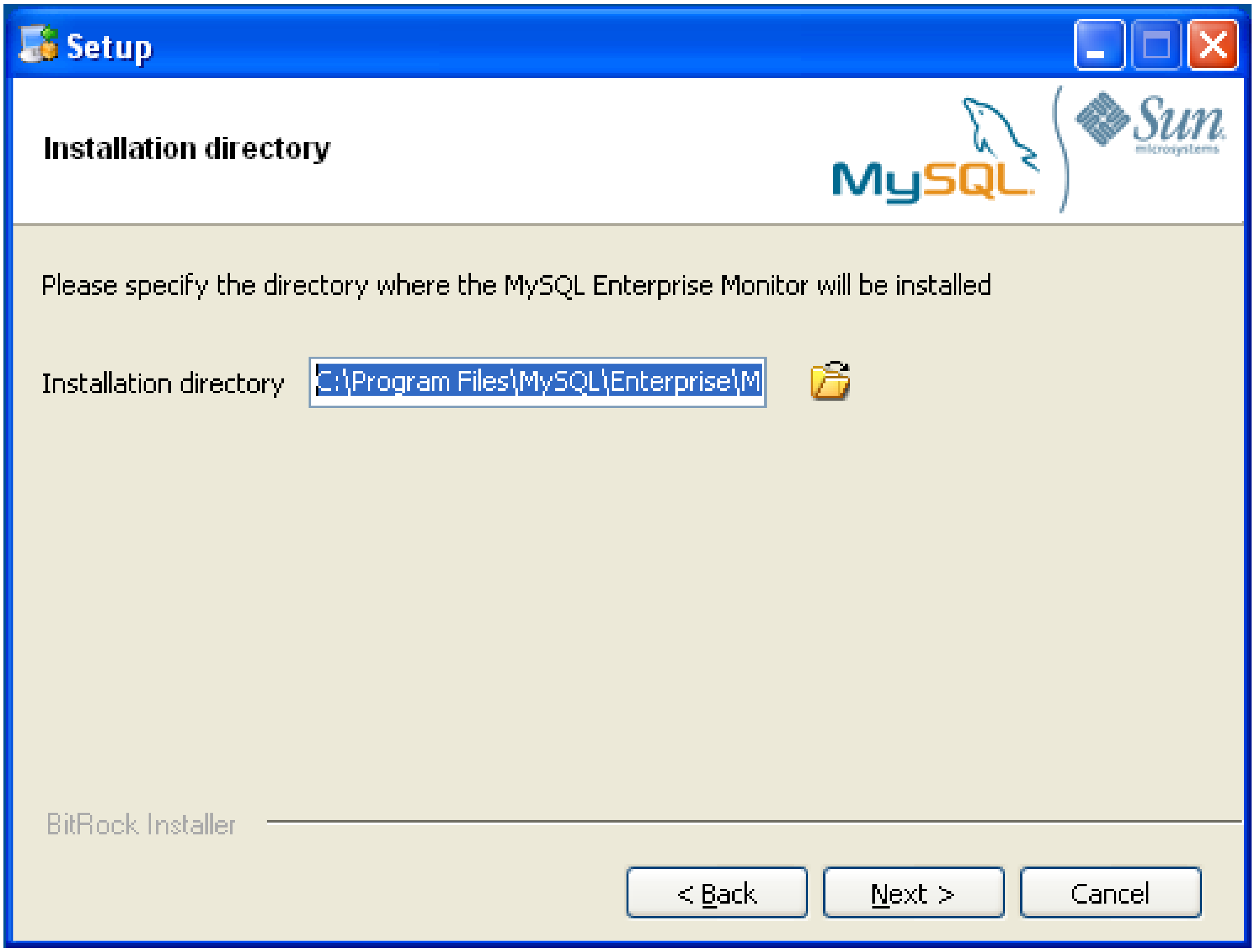
Please select the installation language

English

2. With the installation language selected, the remainder of the installation sets up the installation location and the main configuration parameters required by MySQL Enterprise Service Manager. Click NEXT to continue.

3. Select the installation directory where you want the MySQL Enterprise Service Manager components installed. By default on Windows the directory is `C:\Program Files\MySQL\Enterprise\Monitor`. You click the button next to the installation directory field to select a directory using the File chooser, or type the directory manually. Click NEXT to continue.

Figure 2.2. MySQL Enterprise Monitor: Installing Monitor on Windows: Installation Directory



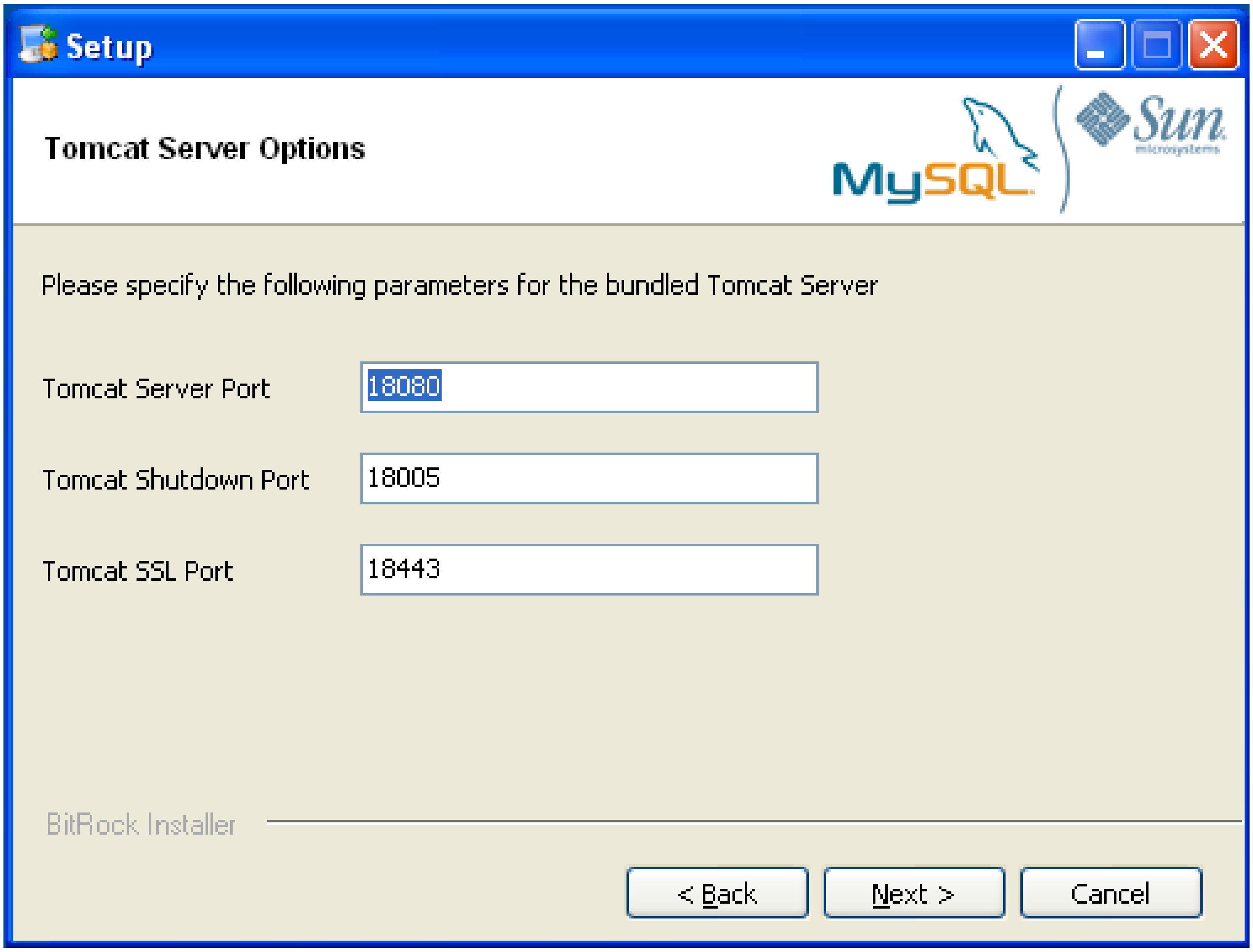
Installation directory

Please specify the directory where the MySQL Enterprise Monitor will be installed

Installation directory C:\Program Files\MySQL\Enterprise\M

4. Configure the options that set the network ports used by the Tomcat server. For more information, see Section 2.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

Figure 2.3. MySQL Enterprise Monitor: Installing Monitor on Windows: Tomcat Server Options



Please specify the following parameters for the bundled Tomcat Server

Tomcat Server Port: 18080

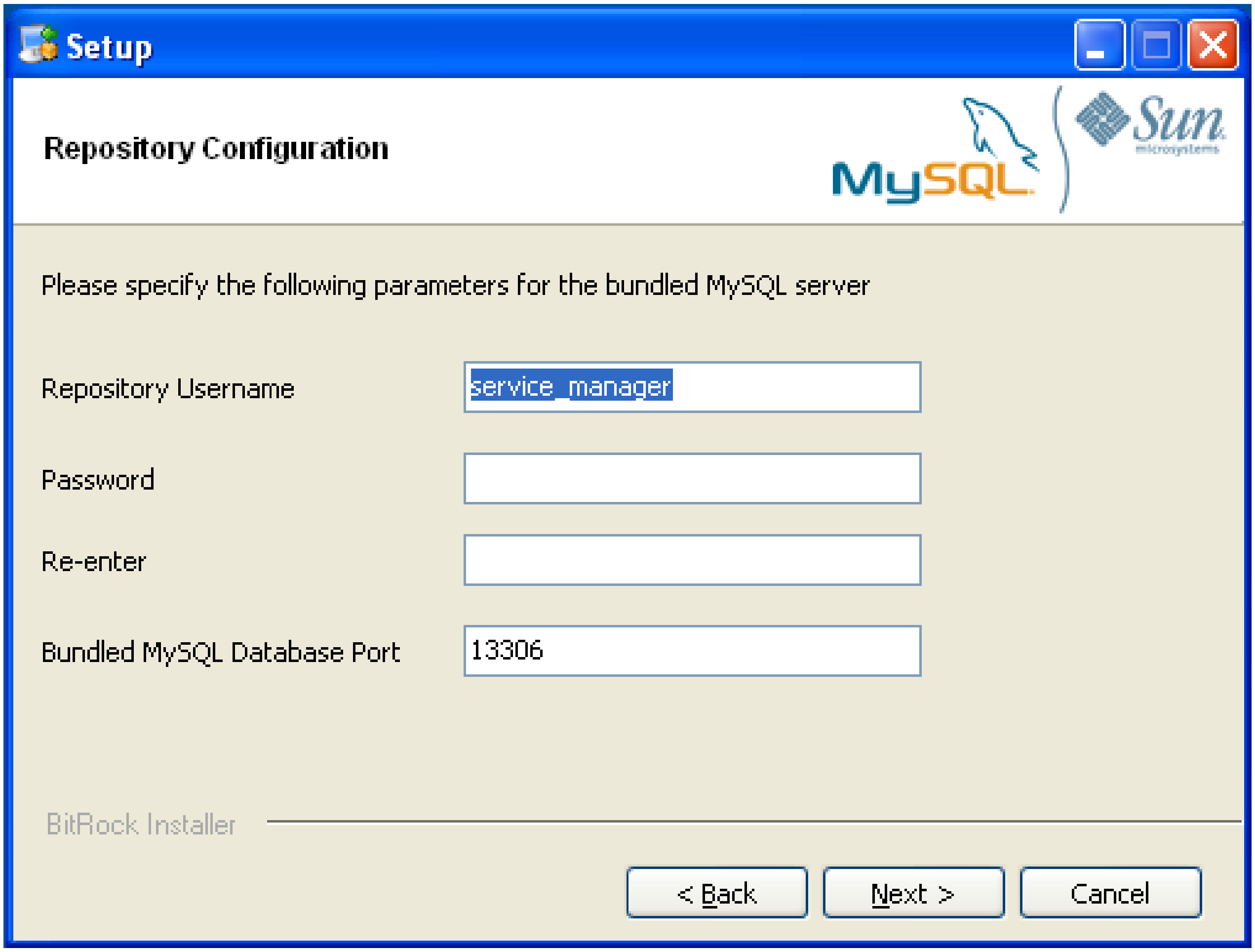
Tomcat Shutdown Port: 18005

Tomcat SSL Port: 18443

5. Configure the repository settings, setting the user name, password and port used to communicate with the bundled MySQL server that will be used to store the information and statistics for your installation. For more information, see Section 2.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

> **Note**
>
> If the Windows firewall is enabled you will be asked to unblock ports for Apache/Tomcat and the MySQL server.

Figure 2.4. MySQL Enterprise Monitor: Installing Monitor on Windows: Repository Configuration



Setup

Repository Configuration

Please specify the following parameters for the bundled MySQL server

Repository Username: service_manager

Password:

Re-enter:

Bundled MySQL Database Port: 13306

6. You will be provided with information and a warning about the configuration options and how they are stored in the `configuration_report.txt` file, and it's location. Take a note of the full path to this file in case you need to look up the information later. Click NEXT to continue.

7. You should now be prompted to start the installation process. Click NEXT to continue.

8. Once the installation has been completed, you will be provided with the information on how to uninstall MySQL Enterprise Service Manager. Click NEXT to continue.

9. To complete the installation and set up your MySQL Enterprise Service Manager, you will need to login to the Dashboard. You can do this automatically by checking the box on the final window before clicking FINISH. This checkbox is selected by default. If you do not want to run the Dashboard at this time, uncheck the box and clock FINISH.

For instructions on starting the MySQL Enterprise Monitor services under Windows, see Section 2.2.5, "Starting/Stopping the MySQL Enterprise Monitor Service on Windows".

2.2.3. Service Manager Installation on Mac OS X

On Mac OS X there are three installation modes `osx`, `text`, and `unattended`. For more information on this topic see Section 2.4, "Unattended Installation". The `text` mode installation for Mac OS X is identical to `text` installation under Unix. For `text` mode installation instructions see Section 2.2.4, "Service Manager Installation on Unix".

Installing the MySQL Enterprise Service Manager on Mac OS X requires an existing installation of Java. The minimum required version is 1.5.0_7. If this version is not installed on your machine you can download it from Apple. This version of Java requires Mac OS X version 10.4.5 as a minimum, so you may need to upgrade your operating system in order to install it.

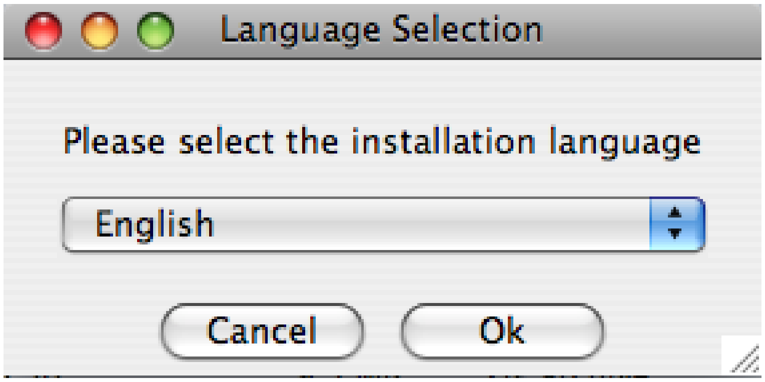
For reasons of backward compatibility, Mac OS X is usually installed with multiple versions of Java. When installing in `osx` mode, version 1.5.0_7 must be the default version. Upon installation, Java 1.5.0_7 sets itself as the default so this is usually not a problem.

If you have changed the default you can reset it or you may install the MySQL Enterprise Service Manager in `text` mode, setting the environment variables to point to the correct version of Java. To install in `text` mode, find the `installbuilder` file in the `Contents/MacOS` directory immediately below the `mysqlmonitor-*version*-osx-installer.app` directory. Installing the MySQL Enterprise Service Manager in `text` mode is identical to the procedure described in Section 2.2.4, “Service Manager Installation on Unix” with the minor differences noted above.

To install using the GUI (`osx`) installation, follow these instructions:

1. Double-click the MySQL Monitor installer. You should be presented with the Language Selection prompt. Select the language to use for the installer and then click OK.

   Figure 2.5. MySQL Enterprise Monitor: Installing Monitor on OS X: Language Selection

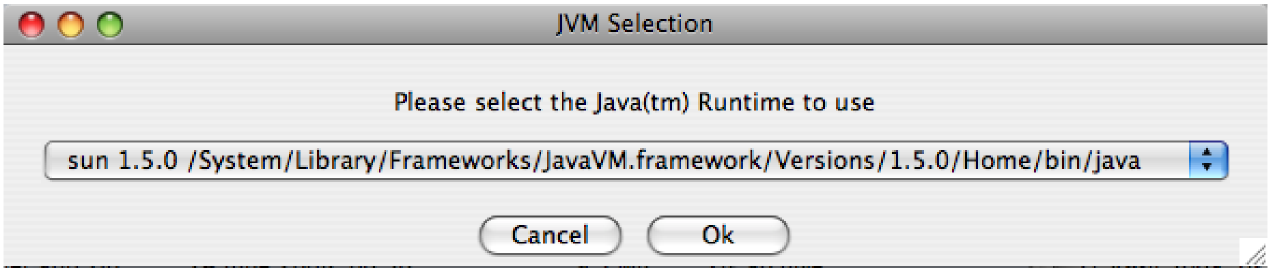
   

   Please select the installation language

   English

2. If you have multiple Java installations on your machine, you will be asked to choose which Java to use with your MySQL Enterprise Service Manager installation. Choose the Java version you want to use (1.5.0 or later is required), and click OK.

   Figure 2.6. MySQL Enterprise Monitor: Installing Monitor on OS X: Java Selection

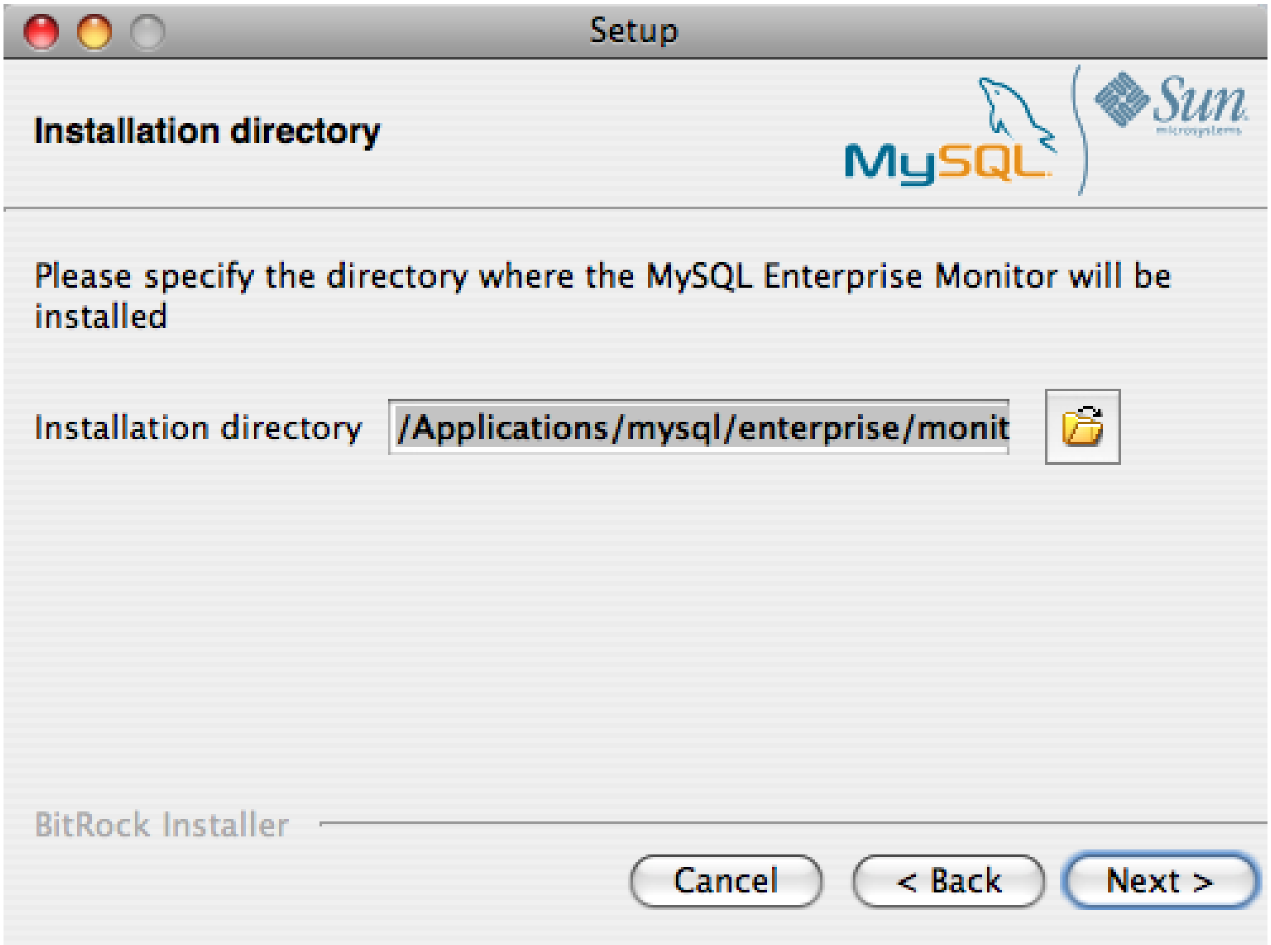
   Please select the Java(tm) Runtime to use

   sun 1.5.0 /System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Home/bin/java

3. With the installation language and Java version selected, the remainder of the installation sets up the installation location and the main configuration parameters required by MySQL Enterprise Service Manager. Click NEXT to contintue.

4. Select the installation directory where you want the MySQL Enterprise Service Manager components installed. By default on Mac OS X the directory is `/Applications/mysql/enterprise/monitor`. You click the button next to the installation directory field to select a directory using the File chooser, or type the directory manually. Click NEXT to continue.

   Figure 2.7. MySQL Enterprise Monitor: Installing Monitor on OS X: Installation Directory



Setup

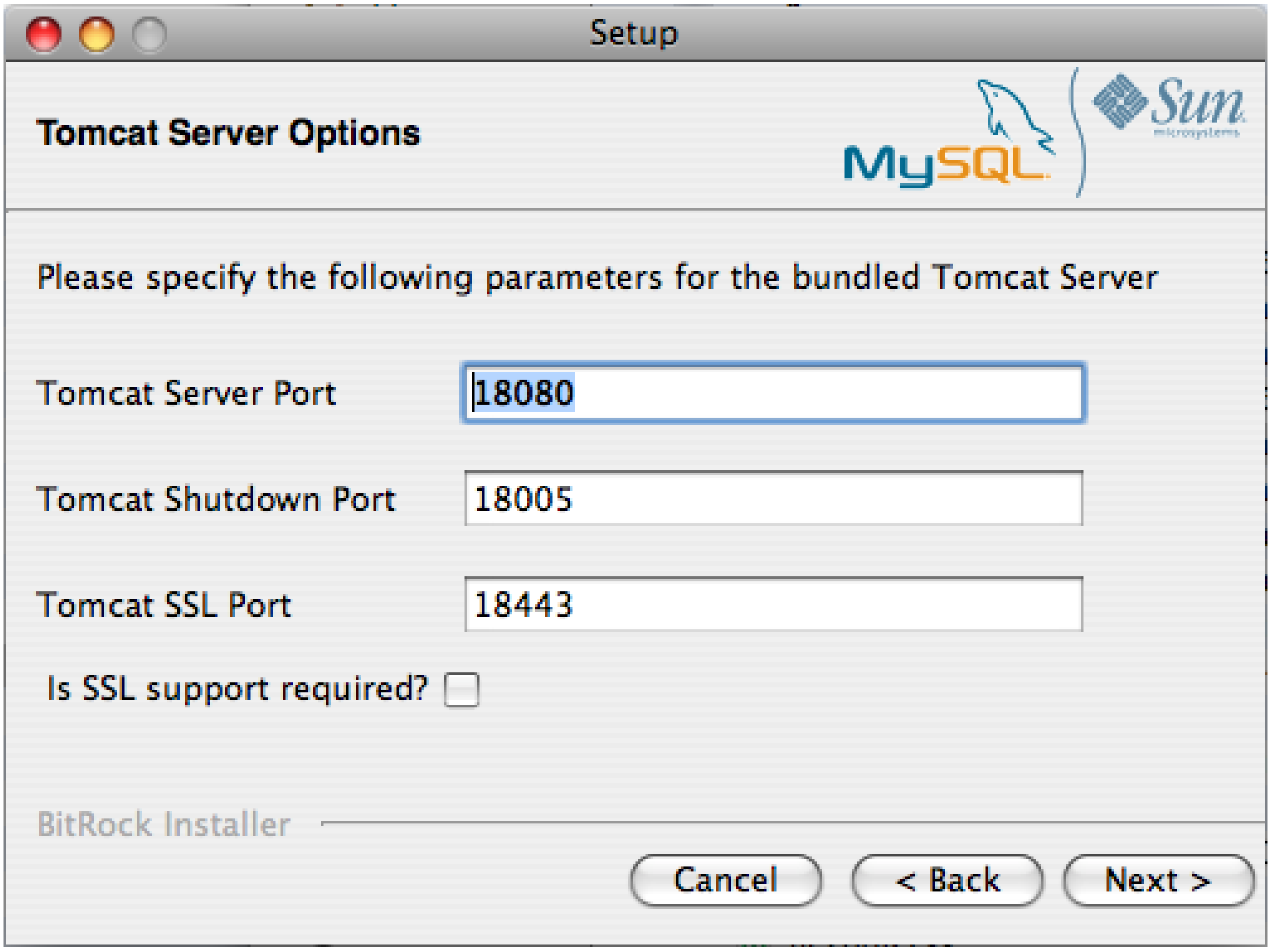
Installation directory

Please specify the directory where the MySQL Enterprise Monitor will be installed

Installation directory /Applications/mysql/enterprise/monit

5. Configure the options that set the network ports used by the Tomcat server. For more information, see Section 2.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

Figure 2.8. MySQL Enterprise Monitor: Installing Monitor on OS X: Tomcat Server Options



Tomcat Server Options

Please specify the following parameters for the bundled Tomcat Server

Tomcat Server Port: 18080

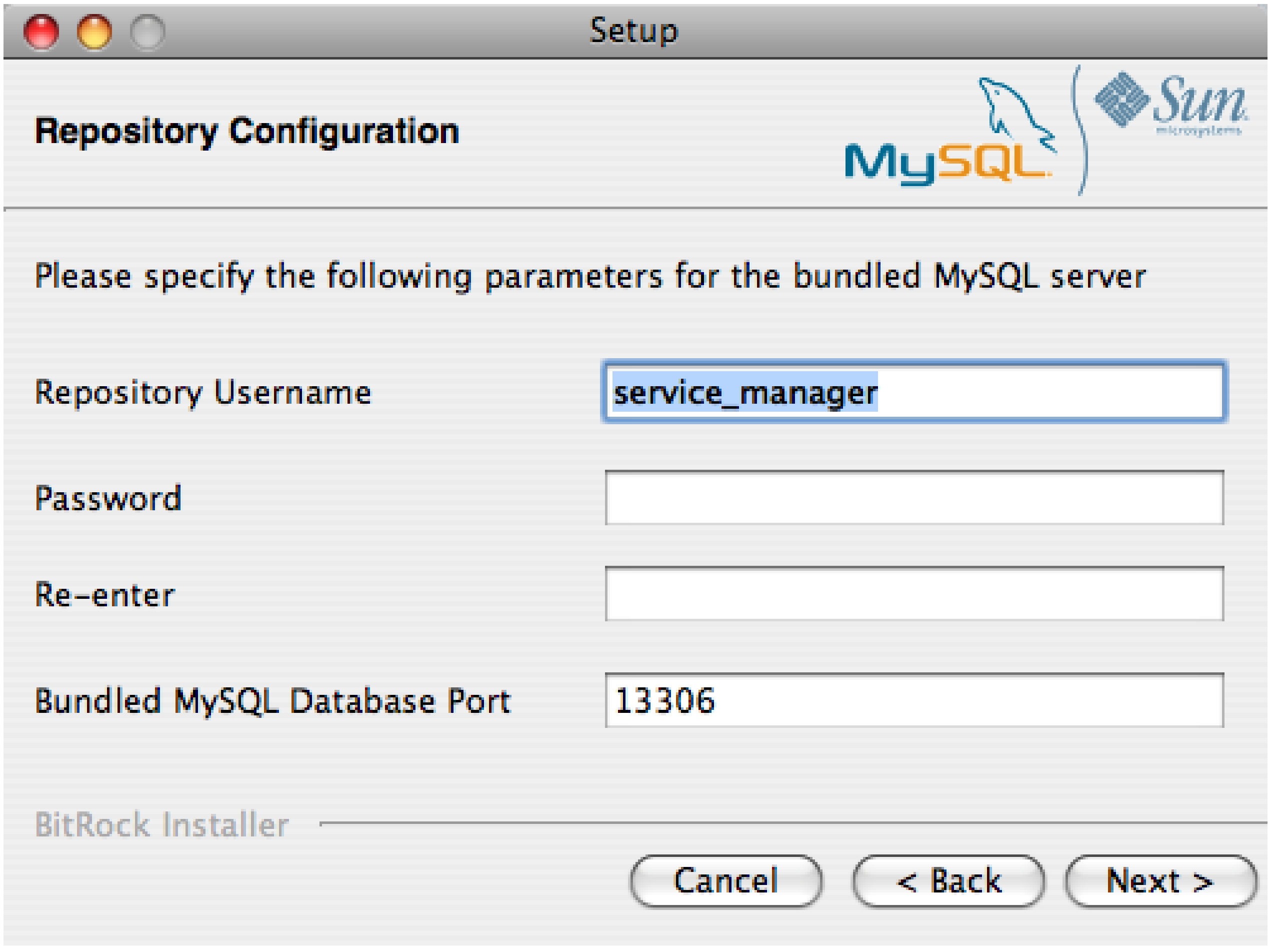
Tomcat Shutdown Port: 18005

Tomcat SSL Port: 18443

Is SSL support required? □

6. Configure the repository settings, setting the user name, password and port used to communicate with the bundled MySQL server that will be used to store the information and statistics for your installation. For more information, see Section 2.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

Figure 2.9. MySQL Enterprise Monitor: Installing Monitor on OS X: Repository Configuration



Repository Configuration

Please specify the following parameters for the bundled MySQL server

Repository Username: service_manager

Password:

Re-enter:

Bundled MySQL Database Port: 13306

7. You will be provided with information and a warning about the configuration options and how they are stored in the `configuration_report.txt` file, and it's location. Take a note of the full path to this file in case you need to look up the information later. Click NEXT to continue.

8. You should now be prompted to start the installation process. Click NEXT to continue.

9. Once the installation has been completed, you will be provided with the information on how to uninstall MySQL Enterprise Service Manager. Click NEXT to continue.

10. To complete the installation and set up your MySQL Enterprise Service Manager, you will need to login to the Dashboard. You can do this automatically by checking the box on the final window before clicking FINISH. This checkbox is selected by default. If you do not want to run the Dashboard at this time, uncheck the box and clock FINISH.

Your installation should now be complete. To continue with the configuration of MySQL Enterprise Service Manager, see Section 2.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

2.2.4. Service Manager Installation on Unix

To install the Service Manager find the file named `mysqlmonitor-version-installer.bin` (where *`version`* indicates the version number, the OS, and the architecture). Ensure that this file is executable by typing:

```
shell> chmod +x mysqlmonitor-version-installer.bin
```

To install to the default directory (`/opt/mysql/enterprise/monitor`) you need to be logged in as `root`. Installing as an unprivileged user installs to the `/home/user_name/mysql/enterprise/monitor/` directory.

What follows describes installation from the command line. You may install the Service Manager graphically by running the installer from within a windows manager. In both cases the steps are identical. You may also install the Service Manager in `unattended` mode. This is especially useful if you are doing multiple installations. For more information on this topic see Section 2.4, "Unattended Installation".

1. Begin installation by typing:

   ```
   shell> ./mysqlmonitor-version-installer.bin
   ```

2. First choose the language for the installation:

   ```
   Language Selection

   Please select the installation language
   [1] English
   [2] Japanese
   Please choose an option [1] :
   ```

3. Throughout the installation process you will be asked the configuration questions for different options. Default values are shown between square brackets; to use the default press **Enter**. Otherwise, enter the new value and press **Enter**:

   First, select the directory where you want MySQL Enterprise Service Manager to be installed. The default is `/opt/mysql/enterprise/monitor/`. Make sure that the location you choose has enough space to hold the installation files and the database information that will be created when MySQL Enterprise Service Manager is running.

   ```

   Please specify the directory where the MySQL Enterprise Service Manager
    will be installed.

   Installation directory [/opt/mysql/enterprise/monitor/]:

   ```

4. Now set the Tomcat Server options. For more details on these parameters, see Section 2.2.1, "Service Manager Installation Common Parameters".

   ```
   ----------------------------------------------------------------------------
   Tomcat Server Options

   Please specify the following parameters for the bundled Tomcat Server

   Tomcat Server Port [18080]:

   Tomcat Shutdown Port [18005]:

   Tomcat SSL Port [18443]:
   ```

   You will also be asked if SSL support is required. SSL support allows your agents and monitor to communicate with each other using SSL. Using SSL means that the data exchanged by the agent and MySQL Enterprise Service Manager are secure and can be used to monitor servers securely, or to monitor agents over a public connection.

   You can enable SSL by pressing **Y** when prompted during installation:

   ```
   Is SSL support required?           [y/N]:
   ```

5. Set the repository (embedded MySQL server) configuration options. For more details on these parameters, see Section 2.2.1, "Service Manager Installation Common Parameters".

   ```
   ----------------------------------------------------------------------------

   Repository Configuration

   Please specify the following parameters for the bundled MySQL server


   Repository Username  [service_manager]:

   Password :

   Re-enter :

   Bundled MySQL Database Port [13306]:

   ```

6. Before the final installation process, you will provided with the location of the file that contains a copy of all of the settings. Be sure to follow the instructions and store this report in a secure location. There is no password recovery feature.

```
----------------------------------------------------------------------------

Configuration Report

Note:

The settings you specified will be saved here:

/opt/mysql/enterprise/monitor/configuration_report.txt

IMPORTANT: This configuration report includes passwords stored in plain text; it
is intended to help you install and configure your agents. We strongly advise
you to secure or delete this text file immediately after installation.


Press [Enter] to continue :
```

7. You you will now be asked to confirm the installation process.

```
Setup is now ready to begin installing MySQL Enterprise Monitor
 on your computer.

Do you want to continue? [Y/n]: Y

Please wait while Setup installs MySQL Enterprise Monitor
 on your computer.
```

   The installation process may take a few minutes to complete. Upon completion you should see:

```
Completed installing files

Setup has completed installing MySQL Enterprise files on your computer

Uninstalling the MySQL Enterprise files can be done by invoking:

/opt/mysql/enterprise/monitor/uninstall

To complete the installation, launch the MySQL Enterprise Dashboard and complete
the initial setup and product activation information. Refer to the readme file
for additional information and a list of known issues.


Press [Enter] to continue :
```

8. Finally, you will be given the opportunity to read a supplied `Readme` file that is supplied with the installation. The `Readme` contains important information about how to use and start your MySQL Enterprise Service Manager.

```
----------------------------------------------------------------------------

Setup has finished installing MySQL Enterprise Monitor on your computer.

View Readme File [Y/n]: n
```

   Once the `Readme` file has been displayed, or if you did not elect to read the file, the installation provides information about how to continue with your installation.

```
Info: To access the MySQL Enterprise Monitor please visit the
following page: http://localhost:18080/Auth.action

Press [Enter] to continue :
```

The Enterprise Dashboard will not start up automatically if you perform a `text` mode installation. For more information on starting and stopping MySQL Enterprise Service Manager, see Section 2.2.6, “Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X”.

2.2.5. Starting/Stopping the MySQL Enterprise Monitor Service on Windows

You can choose to start up the MySQL Enterprise Service Manager on installation. The installed services are called:

- MySQL Enterprise Tomcat
- MySQL Enterprise MySQL

You can stop or start the services from the Microsoft Management Console Services window. Look for the `MySQL Enterprise Tomcat` and the `MySQL Enterprise MySQL` entries.

> **Note**
>
> On Windows Vista or later, starting these services requires administrative privileges—you must be logged in as an administrator. To start or stop a service right-click it and choose the RUN AS ADMINISTRATOR menu option. The same restriction applies to using the menu options discussed in the following and to starting the services from the command line. To open an administrator `cmd` window right-click the `cmd` icon and choose the RUN AS ADMINISTRATOR menu option.

To start or stop a service, right-click it and choose from the options in the pop-up menu.

There is also a menu entry for starting and stopping the services. Navigate to the `Program`, `MySQL`, `MySQL Enterprise Monitor`, `Services` entry to stop or start the services.

You can also stop or start a service from the command line. To start the Tomcat service type:

```
shell> sc start MySQLEnterpriseTomcat
```

or:

```
shell> net start MySQLEnterpriseTomcat
```

To stop this service type:

```
shell> sc stop MySQLEnterpriseTomcat
```

or:

```
shell> net stop MySQLEnterpriseTomcat
```

In similar fashion, you may stop or start the MySQL server from the command line. The service name is `MySQLEnterpriseMySQL`.

You may also start, stop, and restart a specific service or both services using the `mysqlmonitorctl.bat` file. To execute this file, go to the command line and navigate to the `C:\Program Files\MySQL\Enterprise\Monitor` directory. Typing **`mysqlmonitorctl.bat help`** produces the following output:

```
usage: mysqlmonitorctl.bat help
       mysqlmonitorctl.bat (start|stop|restart|install|uninstall)
       mysqlmonitorctl.bat (start|stop|restart) tomcat
       mysqlmonitorctl.bat (start|stop|restart) mysql

help       - this screen
start      - start the service(s)
stop       - stop the service(s)
restart    - restart or start the service(s)
install    - install the service(s)
uninstall  - uninstall the service(s)
```

To stop a specific service, pass the argument `tomcat` or `mysql` in addition to the status change argument. If you wish to change the status of both services, do not specify a service name. You may also uninstall the services using this batch file.

Configuration of the dashboard begins immediately after the Service Manager is installed. To continue a Windows installation skip the next section and go to Section 2.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

2.2.6. Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac

OS X

The services incorporated into the MySQL Enterprise Service Manager are:

- The MySQL Server
- The Apache/Tomcat Server

Should you need to stop, start, or restart the MySQL Enterprise Service Manager call the `mysqlmonitorctl.sh` file located in the `/opt/mysql/enterprise/monitor/` directory on Unix or the `/Applications/mysql/enterprise/monitor/` on Mac OS X. To see all the available options navigate to the appropriate directory and type:

```
shell> /opt/mysql/enterprise/monitor/mysqlmonitorctl.sh help
```

Executing this script produces the following output:

```
usage: ./mysqlmonitorctl.sh help
./mysqlmonitorctl.sh (start|stop|status|restart)
./mysqlmonitorctl.sh (start|stop|status|restart) mysql
./mysqlmonitorctl.sh (start|stop|status|restart) tomcat

help       - this screen
start      - start the service(s)
stop       - stop  the service(s)
restart    - restart or start the service(s)
status     - report the status of the service
```

Using this script you can stop, start, or restart all the Service Manager components. To do this make a call to `mysqlmonitorctl.sh start` from your start-up script.

To start the service:

```
shell> ./mysqlmonitorctl.sh start
./mysqlmonitorctl.sh : mysql  started
nohup: redirecting stderr to stdout
Starting mysqld daemon with databases from /opt/mysql/enterprise/monitor/mysql/data/
Using CATALINA_BASE:   /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_HOME:   /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_TMPDIR: /opt/mysql/enterprise/monitor/apache-tomcat/temp
Using JRE_HOME:       /opt/mysql/enterprise/monitor/java
```

If you try to start the service and it is already running, you will be warned that the services are already running:

```
shell> ./mysqlmonitorctl.sh start
./mysqlmonitorctl.sh : mysql  (pid 18403) already running
./mysqlmonitorctl.sh : tomcat  (pid 18480) already running
```

To stop the service:

```
shell> ./mysqlmonitorctl.sh stop
Using CATALINA_BASE:   /Applications/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_HOME:   /Applications/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_TMPDIR: /Applications/mysql/enterprise/monitor/apache-tomcat/temp
Using JRE_HOME:       /System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Home
Stopping tomcat service .. [ OK ]
STOPPING server from pid file /Applications/mysql/enterprise/monitor/mysql/data/mysqld.pid
090209 15:37:09  mysqld ended
```

The `restart` command is equivalent to executing a `stop` and then `start` operation.

This script can also be used to check the status of the Tomcat web server or the MySQL repository.

```
shell> ./mysqlmonitorctl.sh status
MySQL Network MySQL is running
MySQL Network Tomcat is running
```

Configuration of the dashboard begins immediately after the MySQL Enterprise Service Manager is installed.

2.2.7. MySQL Enterprise Service Manager Configuration Settings and Advisor Installation

The Enterprise Dashboard is the web-based interface to the Service Manager so the procedure for starting the dashboard is identical for all platforms. From the dashboard you can configure the settings necessary for receiving updates from MySQL Enterprise and for the initial installation of the Advisors.

If you installed the Service Manager using a graphical interface, you have the option of launching the dashboard on the final installation screen (as long as the Launch MySQL Enterprise Monitor Now checkbox is checked).

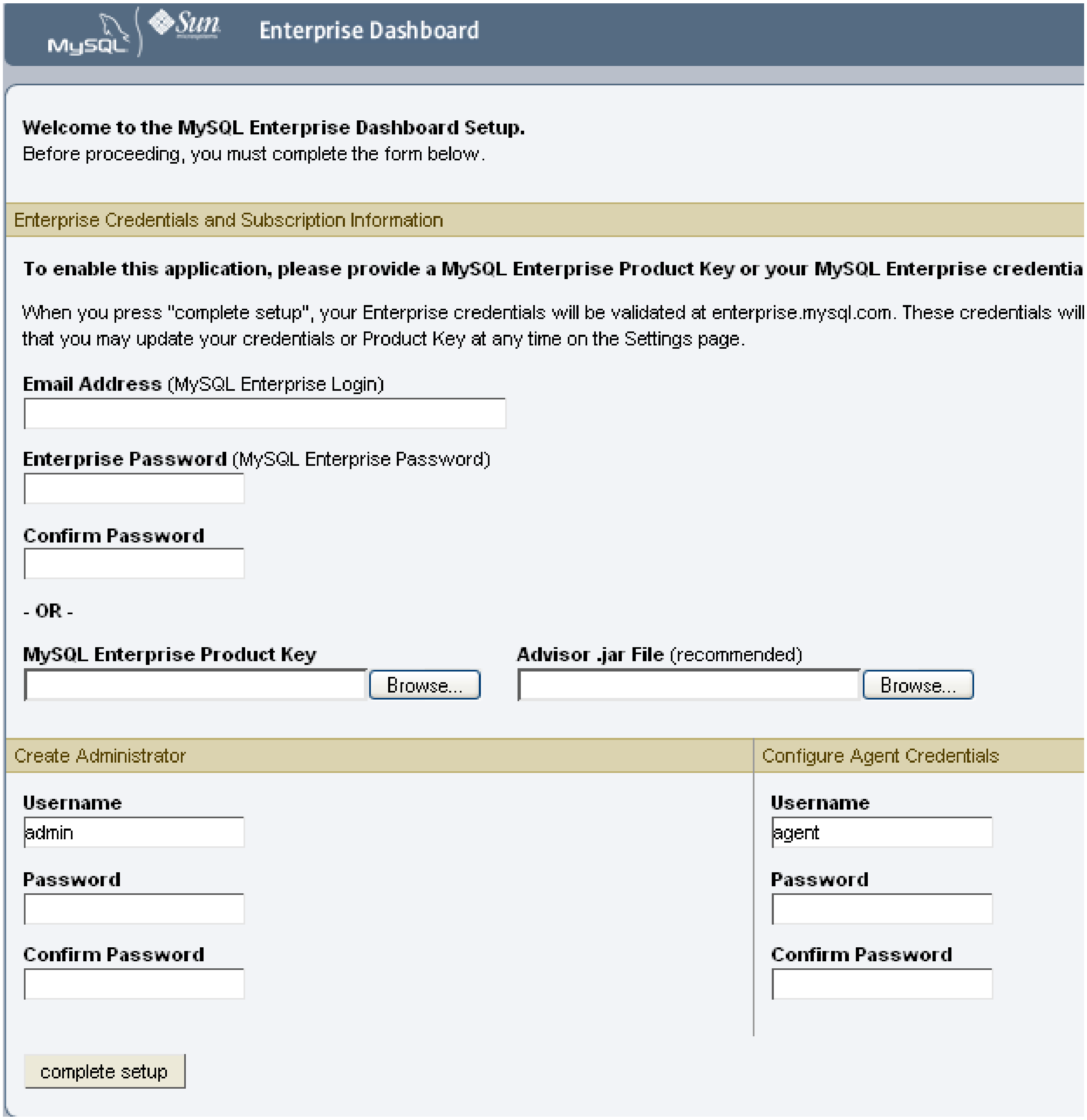
Otherwise, you can view the dashboard by typing `http://localhost:`*`18080`*`/Auth.action` into the address bar of your web browser. If you are unsure of the host name and port to use, check the `configuration_report.txt` file.

Under Windows it is also possible to open the dashboard by choosing the `MySQL` menu item and finding the `MySQL Enterprise Monitor` entry. Under this entry choose `Start Service Manager`.

2.2.7.1. Initial Dashboard Log-In

If this is the first time that you have attempted to log in to the dashboard you should see a screen similar to the following:

Figure 2.10. MySQL Enterprise Monitor: Initial Dashboard Log-In



Welcome to the MySQL Enterprise Dashboard Setup.
Before proceeding, you must complete the form below.

Enterprise Credentials and Subscription Information

To enable this application, please provide a MySQL Enterprise Product Key or your MySQL Enterprise credentia

When you press "complete setup", your Enterprise credentials will be validated at enterprise.mysql.com. These credentials will that you may update your credentials or Product Key at any time on the Settings page.

Email Address (MySQL Enterprise Login)

Enterprise Password (MySQL Enterprise Password)

Confirm Password

- OR -

MySQL Enterprise Product Key Browse...

Advisor .jar File (recommended) Browse...

Create Administrator

Username admin

Password

Confirm Password

Configure Agent Credentials

Username agent

Password

Confirm Password

Use this screen to perform the following tasks:

- Install the Advisors
- Set up your MySQL Enterprise credentials
- Create a user name and password for the dashboard administrator
- Create a user name and password for the Monitor Agent

If you have been provided with a `MySQL Enterprise Product Key` and an Advisors file click the Browse button and locate these files. The advisor file bears the name, `AdvisorScript-`*`version`*`.jar` and the product key, *`Subscription-level_date`*`.xml`. If you do not allow Internet access from the dashboard you must install the advisors in this way. It is strongly recommended that you install the Advisors at this point, but you may do so later. For instructions on doing this see, Section 2.2.7.3, “Installing Advisors After Initial Log-in”. If the product key that you provide is invalid a notification appears and you will be unable to import the advisors.

> **Note**
>
> If you are activating the MySQL Enterprise Monitor using a product key do*not* enter your MySQL credentials; entering both produces an error message.

If you have Internet access from the dashboard, activate MySQL Enterprise Monitor by supplying your MySQL Enterprise credentials. Enter your email address as the `MySQL Enterprise Login` and enter and confirm your MySQL Enterprise password. If you specify incorrect credentials, you receive the error message, “Unable to connect to verify credentials.”

In the **Create Administrator** section of this screen, enter credentials for the dashboard administrator. This creates the `root user` described in Section 2.1.3, “Users Created on First Log-in”. Make note of the user name and password as these credentials are required for any future login.

In the **Configure Agent Credentials** section of this screen enter the credentials for the agent. This is the `agent user` also described in Section 2.1.3, “Users Created on First Log-in”. The agent needs to log in to report its findings. Make note of the agent's credentials; this information is required when installing the agent.

When all the settings have been specified, click the COMPLETE SETUP button. If you log in successfully you should see a message displaying the number of graphs and advisors that have been imported. This number varies depending upon your subscription level.

If importation of the advisor files fails, you will see the message:

```
Unable to import Advisor Jar. You may download the jar
manually from the Enterprise Portal and import it from the 'Check For Updates' page.
```

In this case you may download the advisor file from the Enterprise website and install it as described in Section 2.2.7.3, “Installing Advisors After Initial Log-in”.

2.2.7.2. Setting the Timezone and Locale

If this is the first time that you have launched the dashboard you are asked to set your time zone and locale. Choose the appropriate values from the drop-down list boxes. Setting the time zone ensures that you have an accurate time reference for any notifications from the MySQL Enterprise Advisors.

> **Warning**
>
> It is especially important that the time zone be set correctly as this may also affect the way the graphs display. For this reason, also ensure that the time reported by the operating system is correct. To change the time zone or locale see Section 4.2, “User Preferences”.
>
> The locale chosen determines the user's default language when logging in to the Dashboard. Note that this will override the default browser settings whenever this specific user logs in.

After specifying your time zone and locale, the dashboard opens on the `Monitor` page. For a detailed examination of the `Monitor` Screen see, Chapter 3, *MySQL Enterprise Dashboard*.

2.2.7.3. Installing Advisors After Initial Log-in

The Advisors interpret the data sent by the Monitor Agents and display the results in the dashboard. A minimal set of Advisors are preinstalled with the Service Manager. To obtain the full set of Advisors and get the most value from the MySQL Enterprise Monitor, you must download Advisors from MySQL Enterprise.

If you did not install the Advisors when you first logged in to the MySQL Enterprise Dashboard, open the dashboard, find the `Advisors` tab, and choose the `Check for Updates` link. Doing this downloads the latest version of the Advisors from the MySQL Enterprise web site. In order to install the advisors in this fashion you must specify your MySQL Enterprise credentials. Find instructions for doing this in Section 4.1, “Global Settings”.

If you do not allow Internet access from the dashboard, you must install the Advisors from a local file. To do this you need an advisor file bearing the name, `AdvisorScript-version.jar`. If you don't already have this file, you can find it on the MySQL Enterprise downloads page. Download the Advisors file to a location that is accessible from the dashboard. Use the BROWSE button to find the Advisors file and then choose IMPORT to load the advisors.

2.2.7.4. Upgrading and Updating Advisors

The process for upgrading advisors is exactly the same as the initial installation. Advisors are updated by choosing the UPDATE button on the `Check for Updates` page. If you do not have Internet access from the dashboard you can import the Advisors from a local file as described in Section 2.2.7.3, "Installing Advisors After Initial Log-in".

> **Note**
>
> You may choose to upgrade your MySQL Enterprise Monitor subscription level at any time.

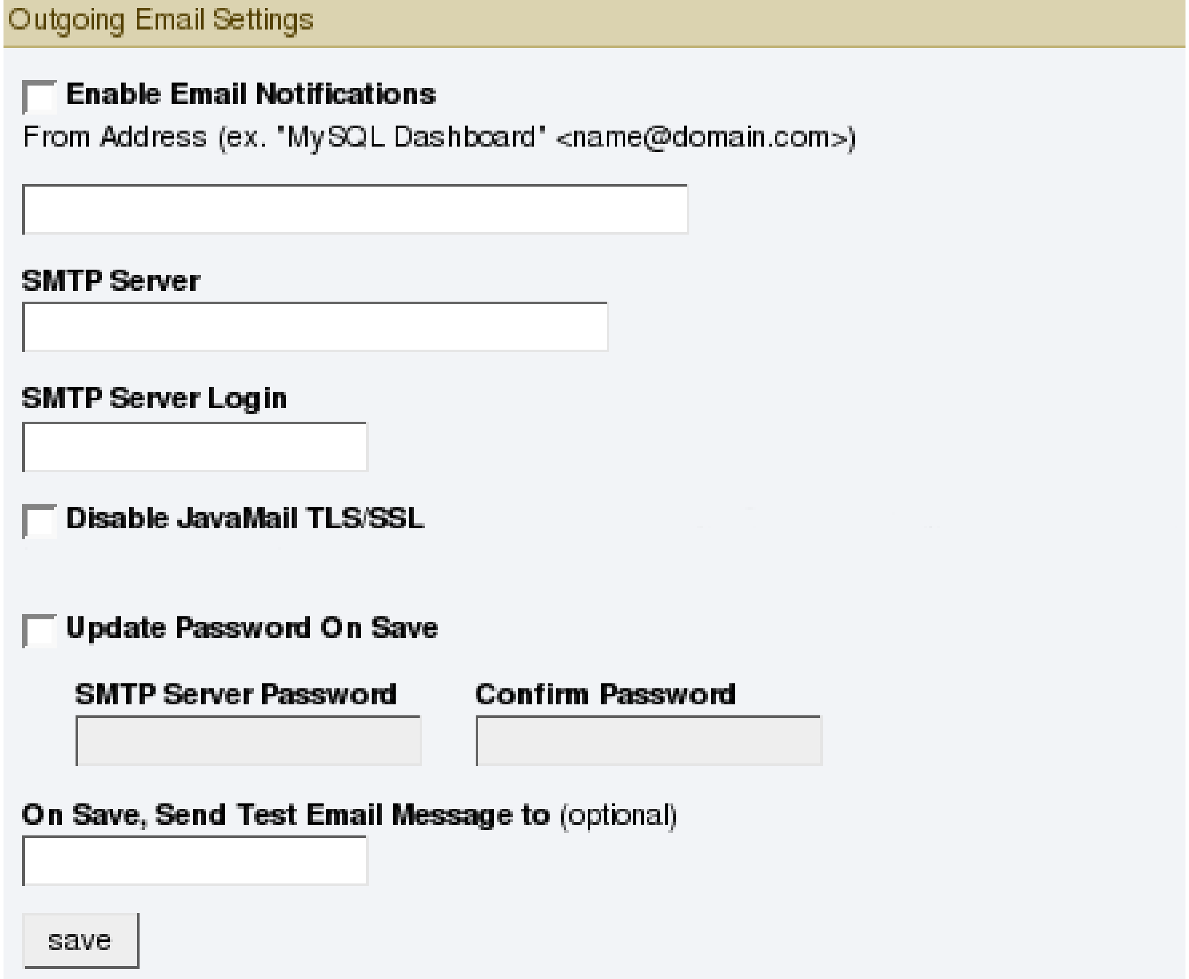
2.2.7.5. Outgoing Email Settings

Alert notification through email is a key component of the MySQL Enterprise Monitor Advisor solution. For this reason you may want to immediately configure an SMTP account for at least one recipient.

To do this choose the `Settings` tab and go to the `Global Settings` screen by clicking the appropriate link. Here you can configure the email settings. These settings apply to the currently logged-in user.

Find the `Outgoing Email Settings` on the left of this page.

Figure 2.11. MySQL Enterprise Monitor: Outgoing Email Settings



Outgoing Email Settings

☐ **Enable Email Notifications**

From Address (ex. "MySQL Dashboard" <name@domain.com>)

SMTP Server

SMTP Server Login

☐ **Disable JavaMail TLS/SSL**

☐ **Update Password On Save**

SMTP Server Password **Confirm Password**

On Save, Send Test Email Message to (optional)

Ensure that the `Enable Email Notifications` checkbox is checked and enter information as appropriate.

The default value for the SMTP port is `25`. If your mail server runs on a different port simply specify it, separating it from the server name using a colon. For example, if your mail server runs on port 587 enter *`email.myserver.com`*`:587` into the **SMTP Server** text box.

> **Note**
>
> An email server must be available for sending email alerts.
>
> The SMTP client uses Transport Layer Security (TLS) if the SMTP server supports it.
>
> If your SMTP server incorrectly indicates that it supports TLS, check the **Disable JavaMail TLS/SSL** check box.

The email settings page is dealt with in more detail in Chapter 4, *The Settings Page*.

2.3. Monitor Agent Installation

A MySQL Enterprise Monitor Agent monitors a MySQL server and sends data to the Advisors. These data are interpreted and displayed in the dashboard. The Monitor Agent is installed on all platforms using the steps described in the next section.

> **Warning**
>
> The MySQL Enterprise Service Manager version 2.0 or higher requires agents with a version number of 2.0 or higher.

2.3.1. Creating a MySQL User Account for the Monitor Agent

Before setting up an agent to monitor a MySQL server you need to ensure that there is a user account for the agent on that server.

The privileges required for this user account vary depending on the information you wish to gather using the MySQL Enterprise Monitor Agent. The following privileges allow the Monitor Agent to perform its assigned duties without limitation:

- `SHOW DATABASES`: Allows the MySQL Enterprise Monitor Agent to gather inventory about the monitored MySQL server.
- `REPLICATION CLIENT`: Allows the MySQL Enterprise Monitor Agent to gather Replication master/slave status data. This privilege is only needed if the MySQL Replication Advisor Rules are employed.
- `SELECT`: Allows the MySQL Enterprise Monitor Agent to collect statistics for table objects.
- `SUPER`: Allows the MySQL Enterprise Monitor Agent to execute `SHOW ENGINE INNODB STATUS` to collect data about InnoDB tables. This privilege is also required to obtain replication information using `SHOW MASTER STATUS`.
- `PROCESS`: When monitoring a MySQL server running MySQL 5.1.24 or above with `InnoDB`, the `PROCESS` privilege is required to execute `SHOW ENGINE INNODB STATUS`.
- `INSERT`: Required to create the UUID required by the agent.
- `CREATE`: Allows the MySQL Enterprise Monitor Agent to create tables. During discovery, the agent creates the table `inventory` within the `mysql` database that is used to the UUID for the server. Without this table, the agent cannot determine the UUID of the server and therefore use this when sending information to MySQL Enterprise Service Manager.

For example, the following `GRANT` statement will give the agent the required `SELECT`, `REPLICATION CLIENT`, `SHOW DATABASES` and `SUPER` rights:

```
GRANT SELECT, REPLICATION CLIENT, SHOW DATABASES, SUPER, PROCESS
  ON *.*
  TO  'mysqluser'@'localhost'
  IDENTIFIED BY 'agent_password';
```

For security reasons, you may wish to limit the `CREATE` and `INSERT` privileges to the agent so that it can only create tables within the `mysql` database:

```
GRANT CREATE, INSERT
  ON mysql.*
  TO  'mysqluser'@'localhost'
  IDENTIFIED BY 'agent_password';
```

To enable replication discovery to work, you should also grant the `SELECT` privilege on the `mysql.inventory` table for each user with replication privileges on the corresponding replication master. This is required to let the MySQL Enterprise Monitor Agent read the replication master UUID. For example:

```
GRANT SELECT
  ON mysql.inventory
  TO  'replicationuser'@'%'
  IDENTIFIED BY 'replication_password';
```

> **Note**
>
> You should perform this step *after* after having run the agent on the corresponding MySQL server to ensure that the `mysql.inventory` table has been correctly created. You can do this by running the agent, shutting the agent down, running the above `GRANT` statement, and then restarting the agent.

If the agent is unable to access the information from the table then a warning containing this information will be written to the agent log.

> **Note**
>
> You may want to disable logging for the grant statement to prevent the grant information being replicated to the slaves. If this is the case, execute the statement `SET SQL_LOG_BIN=0` before you execute the above `GRANT` statement.

In a typical configuration, the agent runs on the same machine as the MySQL server it is monitoring so the host name will be *`localhost`*. However, this will change if the agent is running on a machine other than the one that hosts the monitored MySQL server. In this case, change *`localhost`* to the appropriate value. For more information about remote monitoring see Section 2.3.6.4, “Configuring an Agent to Monitor a Remote MySQL Server”.

2.3.2. Installing the Agent on Microsoft Windows

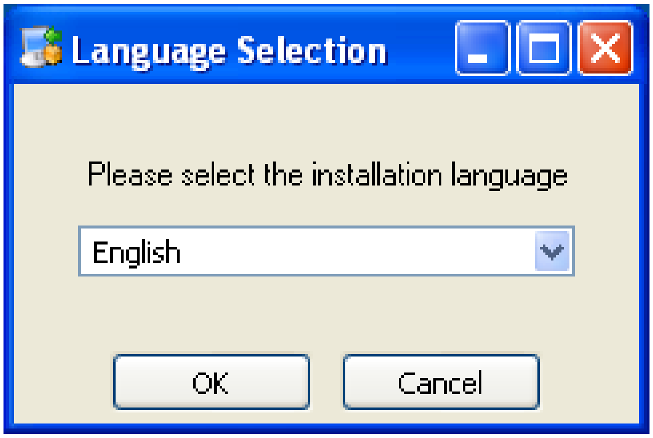
To install the MySQL Enterprise Monitor Agent on Windows, double-click the `mysqlmonitoragent-`*`version`*`-windows-installer.exe` (where *`version`* indicates the three-part version number) installer.

> **Note**
>
> To install the agent as a Windows service, you must do the installation as a privileged user.
>
> On Windows Vista or later, if user account control is on, an operating system dialog box requests confirmation of the installation.

You may also install the Monitor Agent in `unattended` mode. This is especially useful if you are doing multiple installations. For more information on this topic see, Section 2.4, “Unattended Installation”.

1. First, select the language for the MySQL Enterprise Monitor Agent installation. Click OK to continue installation.

   Figure 2.12. MySQL Enterprise Monitor: Installing Agent on Windows: Language Selection



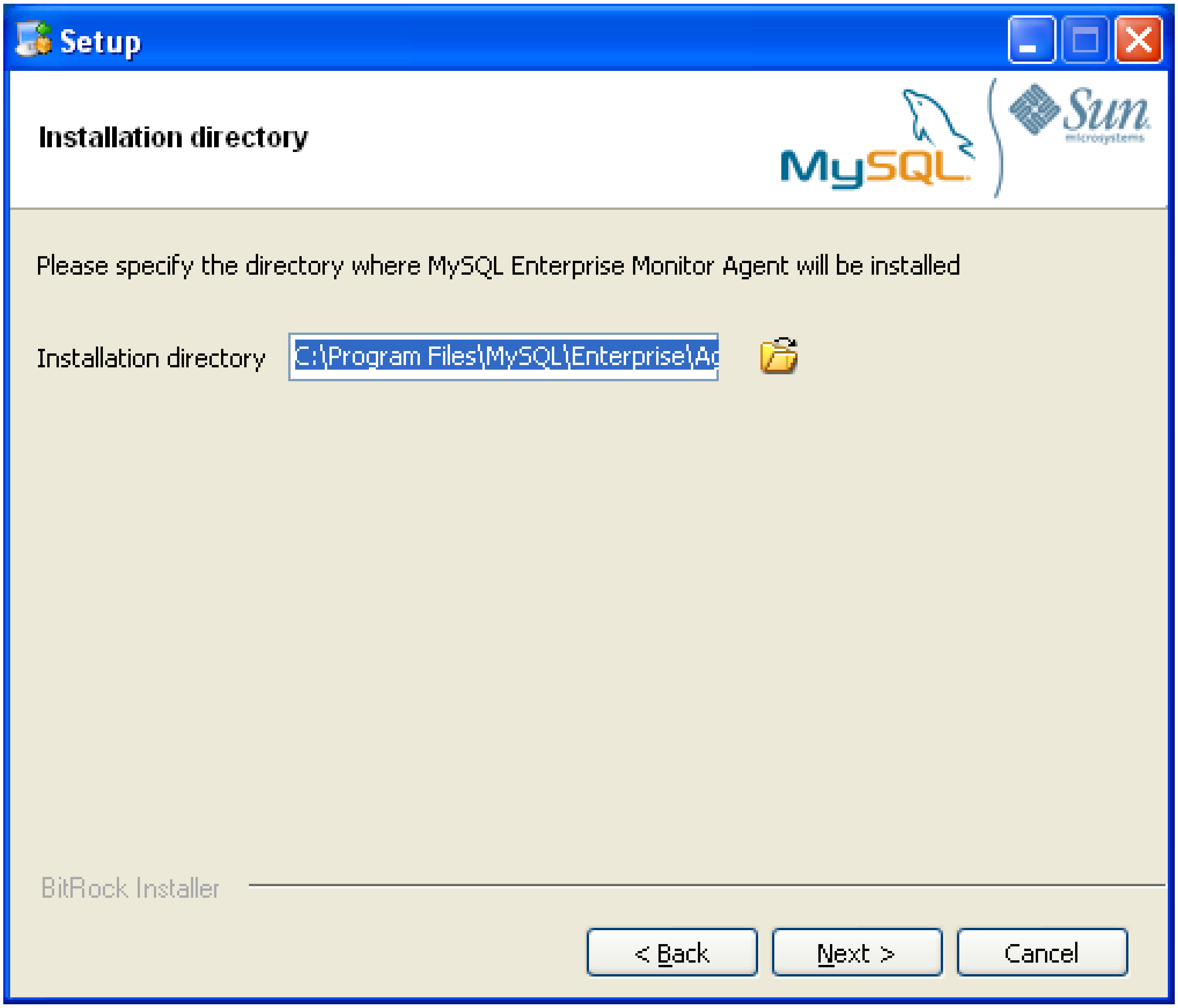
Language Selection

Please select the installation language

English

2. Click NEXT to start the installation process.
3. Select the installation directory. The default installation directory is `C:\Program Files\MySQL\Enterprise\Agent`. Select the installation directory, or type the new directory location. Click NEXT to continue the installation process.

Figure 2.13. MySQL Enterprise Monitor: Installing Agent on Windows: Installation Directory



Setup

Installation directory

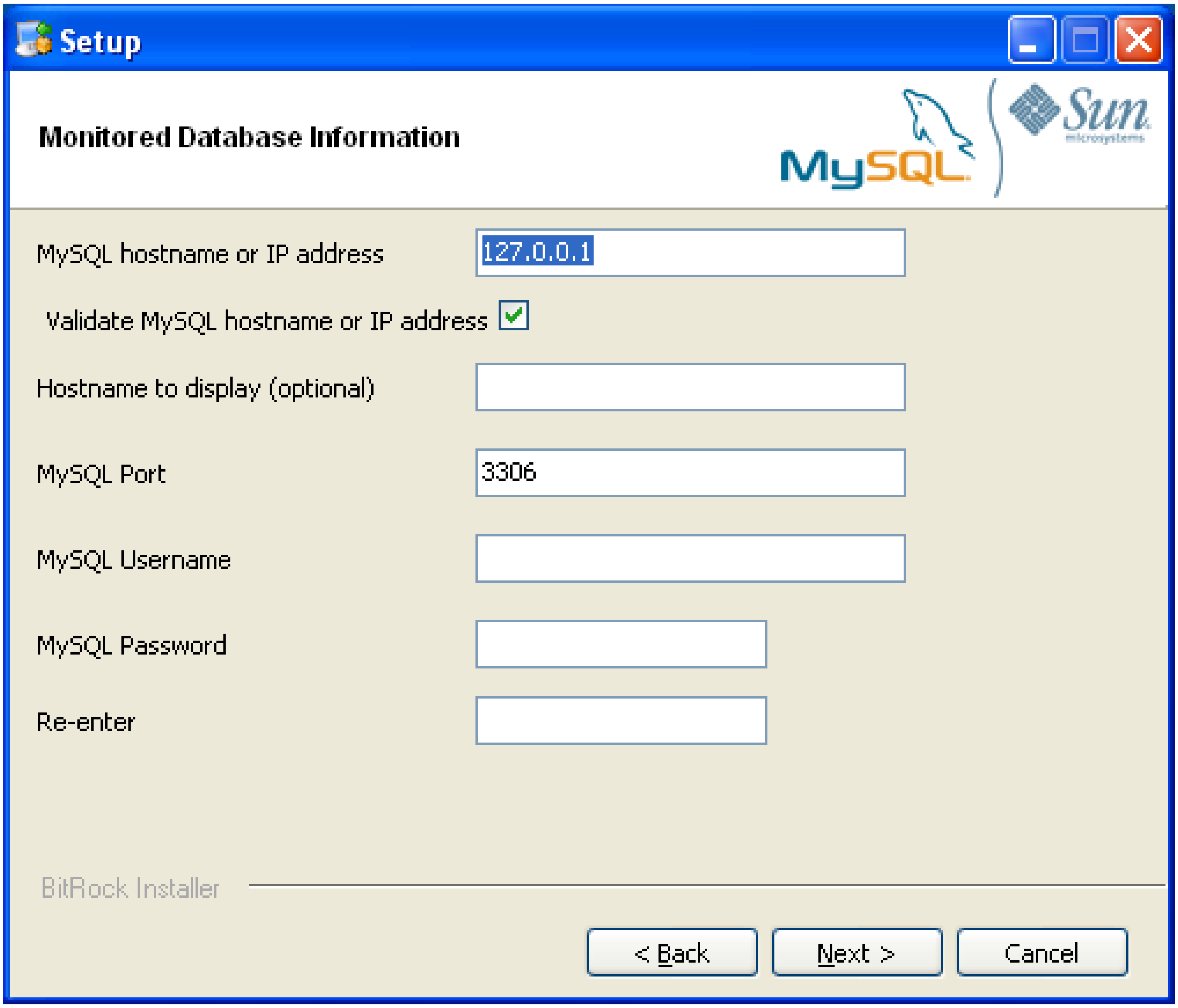
Please specify the directory where MySQL Enterprise Monitor Agent will be installed

Installation directory C:\Program Files\MySQL\Enterprise\Ag

4. You need to specify the information about the MySQL server that you want to monitor. You must enter the IP address or host name of the host you want to monitor, and the port, user name and password that you will use to connect to the MySQL server. If you want to confirm that the MySQL server is currently reachable using the information, ensure that the **Validate MySQL host name or IP address** checkbox is selected.

   > **Note**
   >
   > Currently, on Windows, the monitor agent only includes support for connecting to the server to be monitored using TCP/IP, so if the server has been started with `--skip-networking` it cannot be monitored.
   >
   > If the MySQL server to be monitored has been started using the command option `--bind-address` then the server will only listen for connections on the IP address specified, that is, the IP address of the MySQL server. If the monitor agent has been started using TCP/IP networking and the default address of 127.0.0.1 it will not be able to connect to the server to be monitored. Also, if “localhost” is specified as the host name during agent configuration, a connection will not be established, as the server will be listening for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

   Figure 2.14. MySQL Enterprise Monitor: Installing Agent on Windows: Monitored Database Information



MySQL hostname or IP address: 127.0.0.1

Validate MySQL hostname or IP address ☑

Hostname to display (optional):

MySQL Port: 3306

MySQL Username:

MySQL Password:

Re-enter:

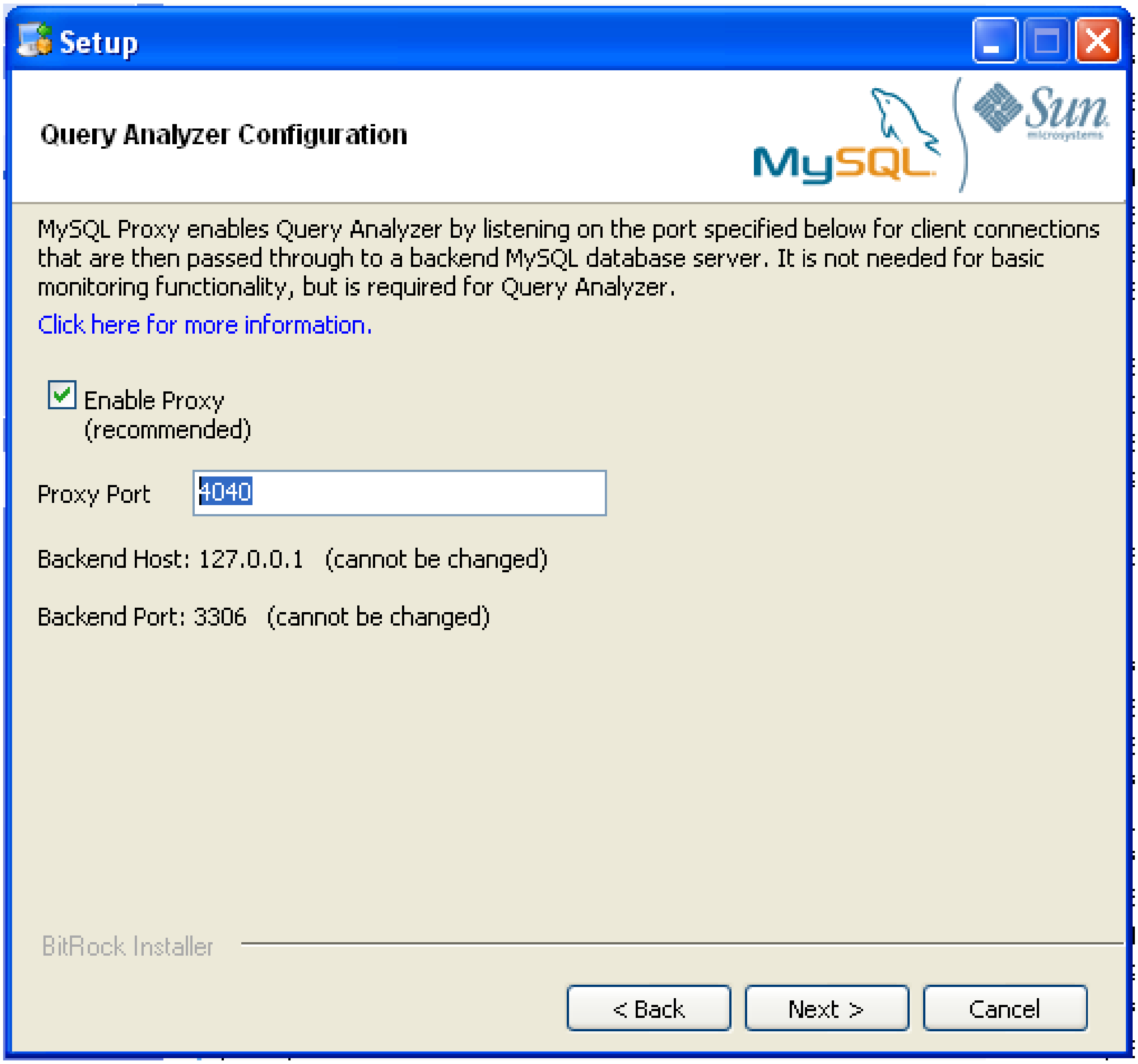
Click NEXT to continue the installation.

5. If you want to use Query Analyzer, then you need to enable the MySQL Enterprise Monitor Agent Proxy. The Proxy is enabled by default. If you disable the Proxy during installation, you will need to enable it later before you are able to use Query Analyzer. For more information on Query Analyzer, see Chapter 8, *The Query Analyzer Page*.

   When Proxy is enabled, MySQL Enterprise Monitor Agent listens on a network port for client applications, and forwards the connections to the backend MySQL server. You can change the port number that MySQL Enterprise Monitor Agent listens for connections.

   The default port is 4040.

Figure 2.15. MySQL Enterprise Monitor: Installing Agent on Windows: Query Analyzer Configuration



Query Analyzer Configuration

MySQL Proxy enables Query Analyzer by listening on the port specified below for client connections that are then passed through to a backend MySQL database server. It is not needed for basic monitoring functionality, but is required for Query Analyzer.

Click here for more information.

☑ Enable Proxy (recommended)

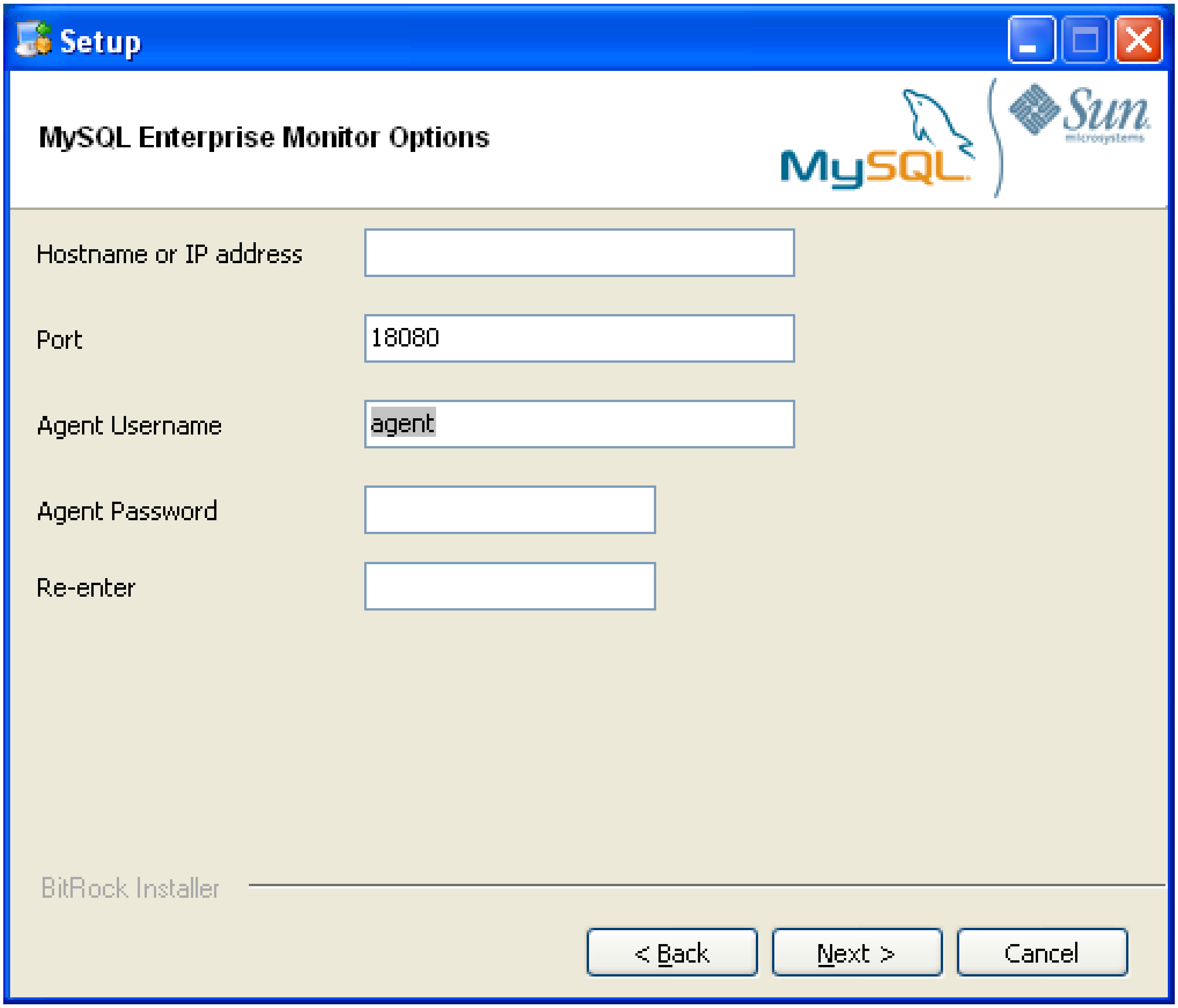
Proxy Port 4040

Backend Host: 127.0.0.1 (cannot be changed)

Backend Port: 3306 (cannot be changed)

6. The MySQL Enterprise Service Manager that you want to use must be configured during installation. The host name, port and agent authentication information must be entered. If you have already installed MySQL Enterprise Service Manager then you can locate the information in the installation report file created during installation. Enter the required information and then click NEXT to continue.

Figure 2.16. MySQL Enterprise Monitor: Installing Agent on Windows: MySQL Enterprise Service Manager Options



MySQL Enterprise Monitor Options

Hostname or IP address

Port 18080

Agent Username agent

Agent Password

Re-enter

7. You will be provided with a Configuration Report containing the information that you have entered during the installation. Check the information provided in the report. If you see a problem, use BACK to go back to the configuration screen and change the information. If the information is correct, click NEXT to continue.

8. You are given a final opportunity to change the installation parameters. Click NEXT to start the installation process.

9. Once the agent has been installed, you will get a confirmation message. Click NEXT to finalize the installation.

10. You can start the MySQL Enterprise Monitor Agent automatically now the installation has been completed. To allow the agent to be started, leave the checkbox selected. To start the agent separately, uncheck the checkbox. Click FINISH to exit the installation.

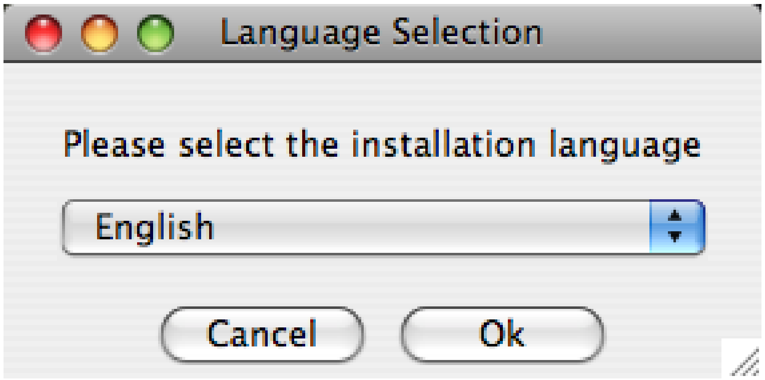
Once the Monitor Agent is installed, it needs to be started. For information on how to start and stop the Agent, see Section 2.3.5.1, “Starting/Stopping the Agent on Windows”.

2.3.3. Installing the Agent on Mac OS X

To install the MySQL Enterprise Monitor Agent on Mac OS X, decompress the `mysqlmonitoragent-*version*-installer.app.zip` and then run the `mysqlenterpriseagent-*version*-installer` application.

1. First, select the language for the MySQL Enterprise Monitor Agent installation. Click OK to continue installation.

   Figure 2.17. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Language Selection

   Please select the installation language

   English

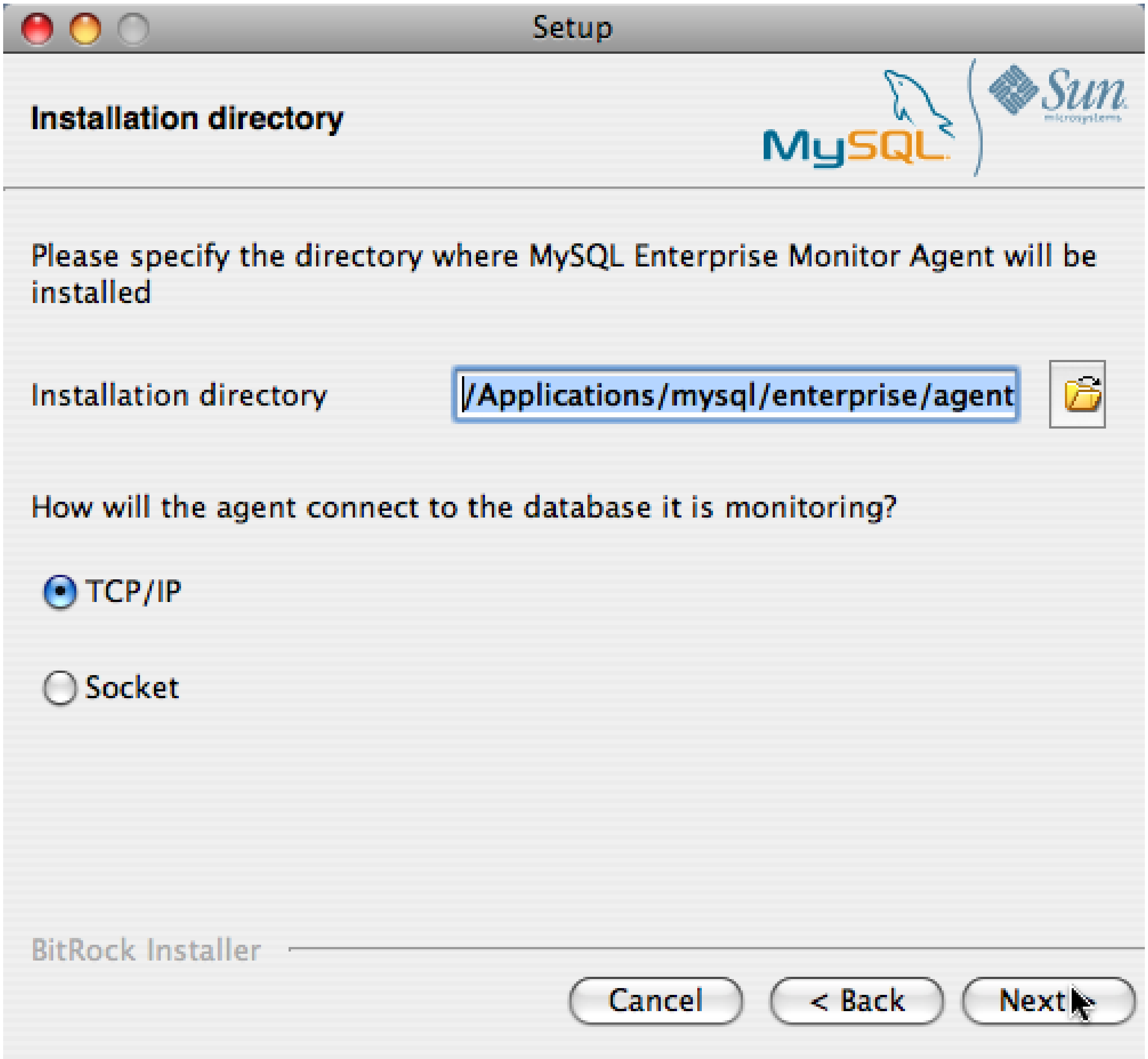
2. Click NEXT to start the installation process.

3. Select the installation directory. The default installation directory is `/Applications/mysql/enterprise/agent`. Select the installation directory, or type the new directory location.

   You also need to select the method that the agent will use to communicate with the MySQL server. You can choose either to use a TCP/IP (network) connection, or a Socket (local) connection. Choose the connection method, and click NEXT.

   > **Note**
   >
   > The monitor agent always associates “localhost” with the TCP/IP address 127.0.0.1, not the MySQL socket. This is in contrast to the MySQL Command Line Tool, which connects using the MySQL socket by default on Unix, if the hostname “localhost” is specified.
   >
   > If the MySQL server you wish to monitor has been started with the `--skip-networking` command option then you will not be able to connect to it using TCP/IP, as the server will not listen for TCP/IP connections. In this case the monitor agent will need to be configured to use the MySQL socket. This can be done during installation by selecting “socket” rather than “TCP/IP” and then specifying the MySQL socket name. This can also be configured after installation by editing the `agent-instance.ini` configuration file, for further information on this refer to Section 2.3.6.2, “MySQL Server (`agent-instance.ini`) Configuration”.
   >
   > If the MySQL server to be monitored has been started using the command option `--bind-address` then the server will only listen for connections on the IP address specified, that is, the IP address of the MySQL server. If the monitor agent has been started using TCP/IP networking and the default address of 127.0.0.1 it will not be able to connect to the server to be monitored. Also, if “localhost” is specified as the host name during agent configuration, a connection will not be established, as the server will be listening for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

   Figure 2.18. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Installation Directory



Installation directory

Please specify the directory where MySQL Enterprise Monitor Agent will be installed

Installation directory: /Applications/mysql/enterprise/agent

How will the agent connect to the database it is monitoring?

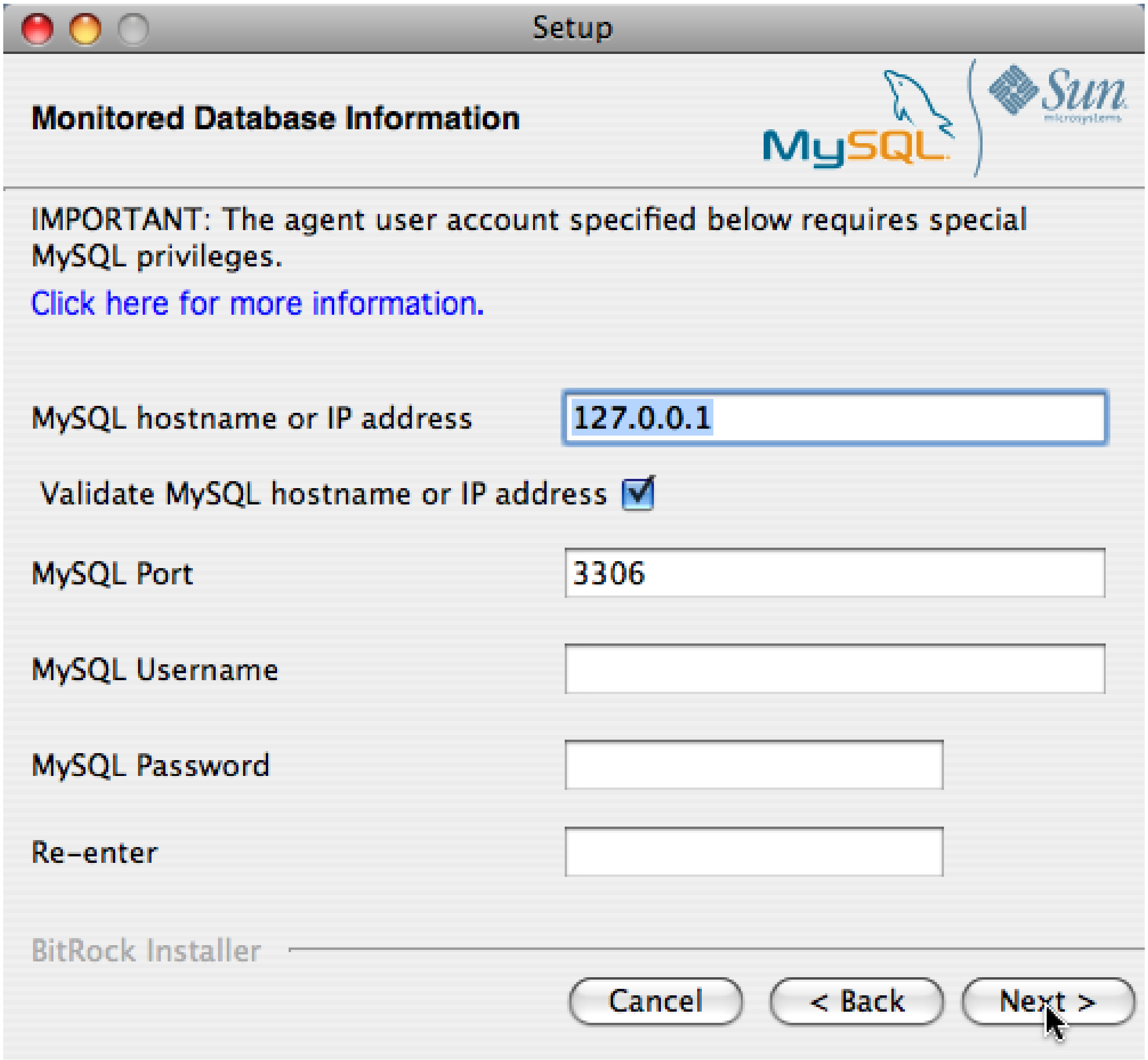
● TCP/IP

○ Socket

4. You need to specify the information about the MySQL server that you want to monitor. The configuration information you enter will depend on the connection method selected in the previous screen.

   - If you chose TCP/IP as the connection method, you must enter the IP address or host name of the host you want to monitor, and the port, user name and password that you will use to connect to the MySQL server. If you want to confirm that the MySQL server is currently reachable using the information, ensure that the **Validate MySQL host name or IP address** checkbox is selected.

     Figure 2.19. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information



Monitored Database Information

IMPORTANT: The agent user account specified below requires special MySQL privileges.

Click here for more information.

MySQL hostname or IP address: 127.0.0.1

Validate MySQL hostname or IP address ☑

MySQL Port: 3306

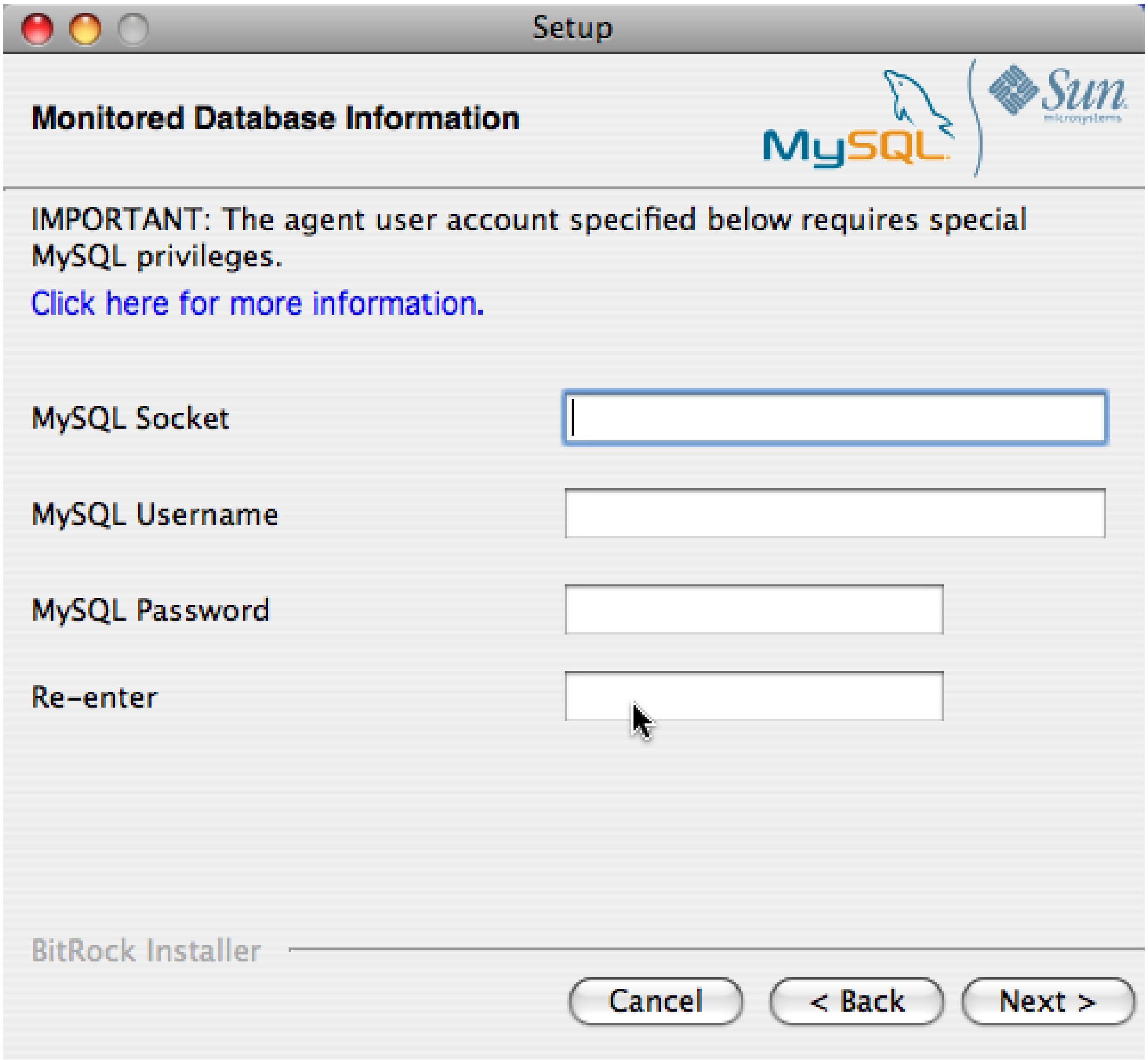
MySQL Username:

MySQL Password:

Re-enter:

- If you chose Socket as the connection method, you must enter the full path name to the Unix socket created by your MySQL server, and the user name and password that will be used to authenticate with the server. Typical values include `/tmp/mysql.sock` and `/var/mysql/mysql.sock`.

Figure 2.20. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information



Monitored Database Information

IMPORTANT: The agent user account specified below requires special MySQL privileges.

Click here for more information.

MySQL Socket

MySQL Username

MySQL Password

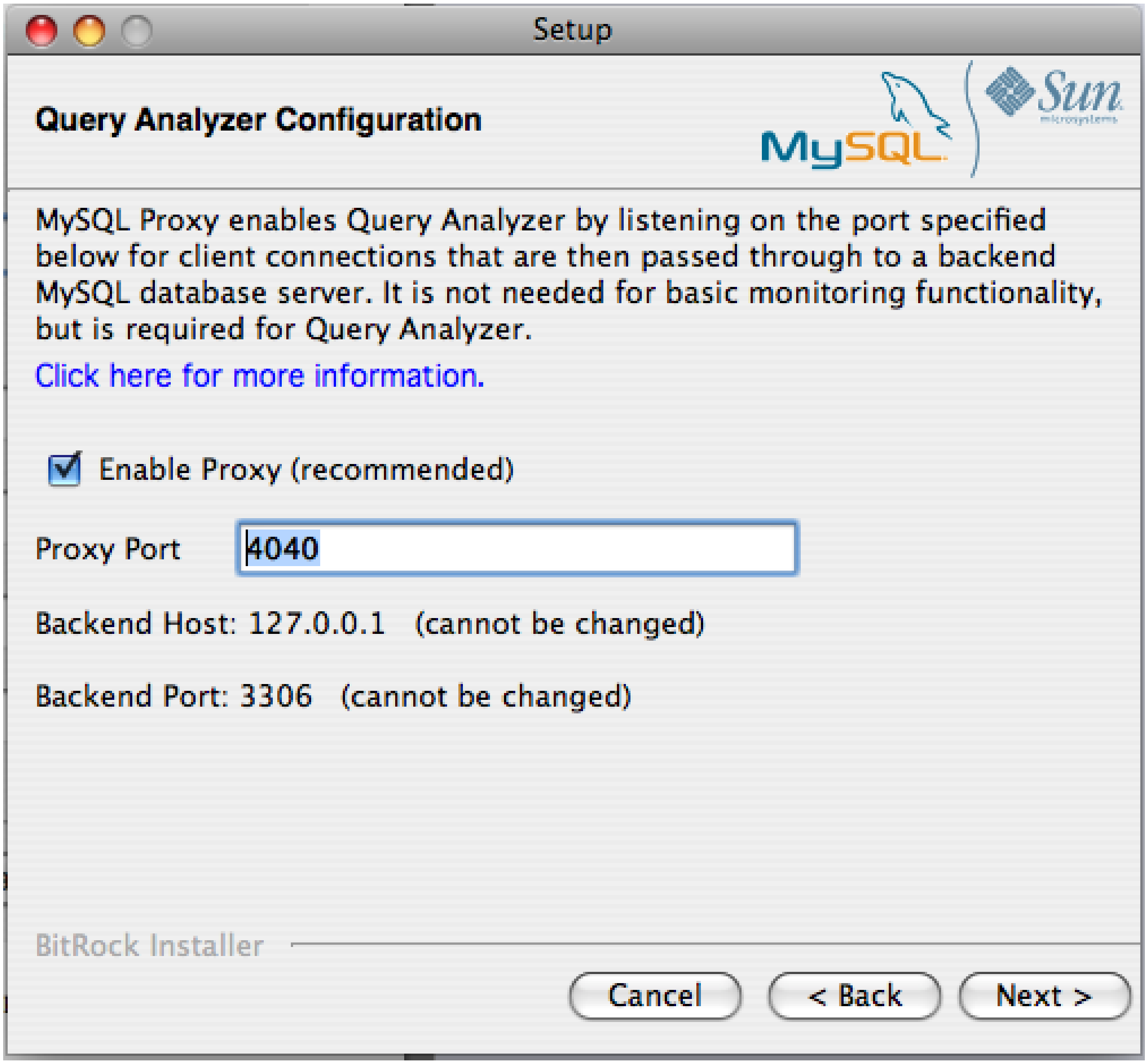
Re-enter

Click NEXT to continue the installation.

5. If you want to use Query Analyzer, then you need to enable the MySQL Enterprise Monitor Agent Proxy. The Proxy is enabled by default. If you disable the Proxy during installation, you will need to enable it later before you are able to use Query Analyzer. For more information on Query Analyzer, see Chapter 8, *The Query Analyzer Page*.

   When Proxy is enabled, MySQL Enterprise Monitor Agent listens on a network port for client applications, and forwards the connections to the backend MySQL server. You can change the port number that MySQL Enterprise Monitor Agent listens for connections The default port is 4040.

Figure 2.21. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Query Analyzer Configuration



Setup

Query Analyzer Configuration

MySQL Proxy enables Query Analyzer by listening on the port specified below for client connections that are then passed through to a backend MySQL database server. It is not needed for basic monitoring functionality, but is required for Query Analyzer.

Click here for more information.

☑ Enable Proxy (recommended)

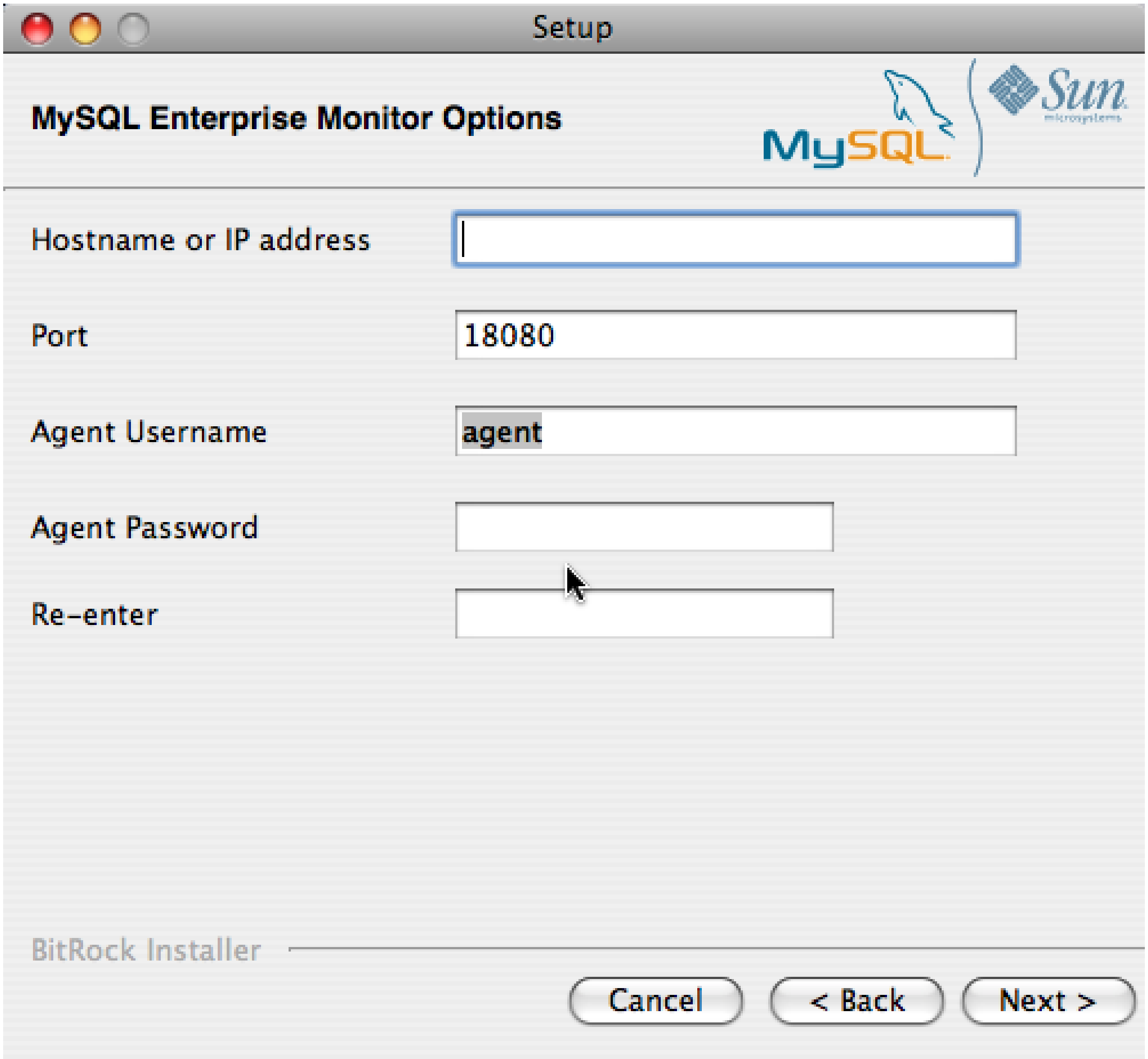
Proxy Port 4040

Backend Host: 127.0.0.1 (cannot be changed)

Backend Port: 3306 (cannot be changed)

6. The MySQL Enterprise Service Manager that you want to use must be configured during installation. The host name, port and agent authentication information must be entered. If you have already installed MySQL Enterprise Service Manager then you can locate the information in the installation report file created during installation. Enter the required information and then click NEXT to continue.

Figure 2.22. MySQL Enterprise Monitor: Installing Agent on Mac OS X: MySQL Enterprise Service Manager Options



MySQL Enterprise Monitor Options

Hostname or IP address

Port 18080

Agent Username agent

Agent Password

Re-enter

7. You will be provided with a Configuration Report containing the information that you have entered during the installation. Check the information provided in the report. If you see a problem, use BACK to go back to the configuration screen and change the information. If the information is correct, click NEXT to continue.

8. You are given a final opportunity to change the installation parameters. Click NEXT to start the installation process.

9. Once the agent has been installed, you will get a confirmation message. Click NEXT to finalize the installation.

10. You can start the MySQL Enterprise Monitor Agent automatically now the installation has been completed. To allow the agent to be started, leave the checkbox selected. To start the agent separately, uncheck the checkbox. Click FINISH to exit the installation.

Once the Monitor Agent is installed, it needs to be started. For information on how to start and stop the Agent, see Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”.

2.3.4. Installing the Monitor Agent on Unix

As a prerequisite for installing the MySQL Enterprise Monitor Agent on Linux systems you must have the Linux Standards Base (LSB) initialization functions installed.

> **Note**
>
> You can check the existence of the LSB components by looking for an LSB package within your Linux package management environment. For example, on RedHat and other RPM-based distributions:
>
> ```
> shell> rpm -qa | grep -i lsb
> redhat-lsb-3.1-19.fc8.x86_64
> ```
>
> Under Debian/Ubuntu:
>
> ```
> shell> dpkg -l|grep -i lsb
> ii  lsb-base                                  3.2-20ubuntu4
>     Linux Standard Base 3.2 init script function
> ii  lsb-release                               3.2-20ubuntu4
>     Linux Standard Base version reporting utilit
> ```
>
> Alternatively, you can use the `lsb_release` command. Existence of this command normally indicates that the current distribution is LSB compliant.

To install the agent navigate to the directory that contains the file, `mysqlmonitoragent-version-installer.bin` (where *`version`* indicates the three-part version number, the OS, and the architecture). Ensure that this file is executable by typing:

```
shell> chmod +x mysqlmonitoragent-version-installer.bin
```

To install to the default directory (`/opt/mysql/enterprise/agent`) you need to be logged in as `root`. Installing as an unprivileged user installs to the `/home/user_name/mysql/enterprise/agent` directory.

> **Note**
>
> If you install the agent as an unprivileged user, it will not automatically start up on rebooting.

What follows describes installation from the command line. You may install the Monitor Agent graphically by running the installer from within a windows manager. In both cases the steps are identical. You may also install the Monitor Agent in `unattended` mode. This is especially useful if you are doing multiple installations. For more information on this topic see Section 2.4, “Unattended Installation”.

Begin installation from the command line by typing:

```
shell> ./mysqlmonitoragent-version-installer.bin --mode text
```

The various options are shown in what follows. Default values are indicated by square brackets; to select them press ENTER. Otherwise enter a value of your choosing.

1. First, you must select the Language you want to use during the installation process:

   ```
   Language Selection

   Please select the installation language
   [1] English
   [2] Japanese
   Please choose an option [1] :
   ```

2. Next, specify the directory where you want the agent installed:

   ```
   ----------------------------------------------------------------------------
   Welcome to the MySQL Enterprise Monitor Agent Setup Wizard.

   ----------------------------------------------------------------------------
   Please specify the directory where MySQL Enterprise Monitor Agent will be installed

   Installation directory [/opt/mysql/enterprise/agent]:
   ```

3. Specify the MySQL server that you want to monitor. First, you must specify whether you want to use a TCP/IP or socket-based connection to communicate with the MySQL Server:

```
How will the agent connect to the database it is monitoring?


[1] TCP/IP
[2] Socket
Please choose an option [1] :
```

If you select TCP/IP, then you will be asked to enter the TCP/IP address and port number:

```
----------------------------------------------------------------------------
Monitored Database Information

IMPORTANT: The agent user account specified below requires special MySQL privileges.

Visit the following URL for more information:
https://enterprise.mysql.com/docs/monitor/2.0/en/mem-install.html#mem-agent-rights

MySQL hostname or IP address [127.0.0.1]:



Validate MySQL hostname or IP address [Y/n]:


MySQL Port [3306]:
```

If you select Socket, then you will be asked to provide the path name to the MySQL socket. Typical values are `/tmp/mysql.sock`, `/var/lib/mysql.sock`, or `/var/run/mysql.sock`.

```
----------------------------------------------------------------------------
Monitored Database Information

IMPORTANT: The agent user account specified below requires special MySQL privileges.

Visit the following URL for more information:
https://enterprise.mysql.com/docs/monitor/2.0/en/mem-install.html#mem-agent-rights

MySQL Socket []:
```

> ### Note
>
> The monitor agent always associates “localhost” with the TCP/IP address 127.0.0.1, not the MySQL socket. This is in contrast to the MySQL Command Line Tool, which connects using the MySQL socket by default on Unix, if the hostname “localhost” is specified.
>
> If the MySQL server you wish to monitor has been started with the `--skip-networking` command option then you will not be able to connect to it using TCP/IP, as the server will not listen for TCP/IP connections. In this case the monitor agent will need to be configured to use the MySQL socket. This can be done during installation by selecting “socket” rather than “TCP/IP” and then specifying the MySQL socket name. This can also be configured after installation by editing the `agent-instance.ini` configuration file, for further information on this refer to Section 2.3.6.2, “MySQL Server (`agent-instance.ini`) Configuration”.
>
> If the MySQL server to be monitored has been started using the command option `--bind-address` then the server will only listen for connections on the IP address specified, that is, the IP address of the MySQL server. If the monitor agent has been started using TCP/IP networking and the default address of 127.0.0.1 it will not be able to connect to the server to be monitored. Also, if “localhost” is specified as the host name during agent configuration, a connection will not be established, as the server will be listening for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

4. Specify the user credentials for the MySQL server that you want to monitor:

```
MySQL Username []: service_agent

MySQL Password :

Re-enter :
```

5. Select whether you want to enable Query Analyzer. If you disable the Query Analyzer during installation, you will need to manually edit the configuration file to re-enable the Query Analyzer functionality. If you enable Query Analyzer (Proxy), you must specify the port on which the agent will listen for queries.

```
----------------------------------------------------------------------------
Query Analyzer Configuration

MySQL Proxy enables query monitoring and analysis by listening on a specified port for client connections that are the

Click here for more information.
 [Y/n]:

Enable Proxy (recommended) [Y/n]:


Proxy Port [4040]:

Backend Host: 127.0.0.1   (cannot be changed)

Backend Port: 3306   (cannot be changed)
```

For more information on enabling Query Analyzer if you disabled it during installation, see Chapter 8, *The Query Analyzer Page*.

6. Enter the details of the MySQL Enterprise Service Manager that you want to use with this agent. The configuration information required is available within the installation report generated when you installed MySQL Enterprise Service Manager

   ```

   ----------------------------------------------------------------------------

   MySQL Enterprise Monitor Options


   Hostname or IP address []: 192.168.0.197

   Tomcat Server Port [18080]:

   Tomcat SSL Port [18443]:
   ```

   The agent and MySQL Enterprise Service Manager support using SSL for communication. If you want to enable SSL communication between the agent and the MySQL Enterprise Service Manager, you must reply **Y** to the following question.

   ```

   Use SSL? [y/N]:

   Agent Username [agent]:

   Agent Password :

   Re-enter :

   ----------------------------------------------------------------------------

   ```

7. Before installation starts, you will be provided with a summary of the installation settings that you have specified:

   ```
   Here are the settings you specified:

   Installation directory: /opt/mysql/enterprise/agent

   Monitored MySQL Database:
   -------------------------
   Hostname or IP address: 127.0.0.1
   Port: 3306
   MySQL username: mysql_user
   MySQL password: password

   Query Analyzer Configuration
   -------------------------
   Proxy Enabled: yes
   Proxy Port: 4040

   MySQL Enterprise Manager:
   ------------------------------

   Hostname or IP address: 192.168.0.197

   Tomcat Server Port: 18080
   Tomcat SSL Port: 18443
   Use SSL: 0
   Agent username: agent

   Press [Enter] to continue :


   ----------------------------------------------------------------------------
   ```

```
Setup is now ready to begin installing MySQL Enterprise Monitor Agent on your computer.

Do you want to continue? [Y/n]: y
```

8. The installer will copy the necessary files and create the configuration file required to run the agent:

```
----------------------------------------------------------------------------
Please wait while Setup installs MySQL Enterprise Monitor Agent on your computer.

 Installing
 0% ______________ 50% ______________ 100%
 #########################################

----------------------------------------------------------------------------
Info to start MySQL Agent

The MySQL agent was successfully installed. To start the MySQL Agent please
invoke:
/opt/mysql/enterprise/agent/etc/init.d/mysql-monitor-agent start
Press [Enter] to continue :

----------------------------------------------------------------------------
Setup has finished installing MySQL Enterprise Monitor Agent on your computer.
```

9. Finally, you can read the supplied `README` file when prompted. The file is provided within the `share/doc/README_en.txt` file within the agent installation directory if you would like to read this file separately.

For information on starting the agent, see Section 2.3.5.3, “Starting/Stopping the Agent on Unix”.

2.3.5. Starting/Stopping the MySQL Enterprise Monitor Agent

The MySQL Enterprise Monitor Agent can be started and stopped at any time. When not running, information about the current status of your server will not be available, and MySQL Enterprise Service Manager will provide a warning if an agent and the MySQL server that it monitors is unavailable.

> **Warning**
>
> If you are using Query Analyzer, then turning off the agent will prevent your applications from communicating with the MySQL server. See Chapter 8, *The Query Analyzer Page*.

2.3.5.1. Starting/Stopping the Agent on Windows

You have the option of starting the Monitor Agent from the final installation screen. Otherwise you can do this by going to the `Start Menu` and under `Programs` find `MySQL` and then the `MySQL Enterprise Monitor Agent` entry. Simply select the `Start MySQL Enterprise Monitor Agent` option.

> **Note**
>
> On Windows Vista or later, starting the agent requires administrative privileges—you must be logged in as an administrator. To start or stop the agent right-click the menu item and choose the Run as Administrator menu option. The same restriction applies to starting the agent from the command line. To open an administrator `cmd` window right-click the `cmd` icon and choose the Run as Administrator menu option.

> **Warning**
>
> To report its findings, the agent needs to be able to connect to the dashboard through the port specified during installation. The default value for this port is `18080`; ensure that this port is not blocked. If you need help troubleshooting the agent installation see, Section 2.3.7, “Troubleshooting the Agent”.

Alternately, you can start the agent from the command line by entering:

```
shell> sc start MySQLEnterpriseMonitorAgent
```

or:

```
shell> net start MySQLEnterpriseMonitorAgent
```

You can also start the agent by issuing the command, `agentctl.bat start`. Stop the agent by passing the argument, `stop`. This batch file is found in the `Agent` directory.

For confirmation that the service is running you can open the Microsoft Management Console Services window. To do this go to the Control Panel, find `Administrative Tools` and click the link to `Services`. Locate the service named `MySQL Enterprise Monitor Agent` and look under the **Status** column.

You may also start the agent from this window rather than from the `Start` menu or the command line. Simply right-click `MySQL Enterprise Monitor Agent` and choose `Start` from the pop-up menu. Starting the agent from this window opens an error dialog box if the agent cannot connect to the MySQL server it is monitoring. No error is displayed if the agent is unable to connect to the MySQL Enterprise Service Manager.

The pop-up menu for starting the agent also offers the option of stopping the agent. To stop the agent from the command line you only need type:

```
shell> sc stop MySQLEnterpriseMonitorAgent
```

or:

```
shell> net stop MySQLEnterpriseMonitorAgent
```

> **Note**
>
> `MySQLEnterpriseMonitorAgent` is the default name of the Monitor Agent service. If you have added an additional agent as described in Section 2.3.6.2, “MySQL Server (`agent-instance.ini`) Configuration”, replace `MySQLEnterpriseMonitorAgent` with the appropriate agent name.

2.3.5.2. Starting/Stopping the Agent on Mac OS X

The script to start the agent on Mac OS X is located in the `/Applications/mysql/enterprise/agent/etc/init.d` directory. To start the agent navigate to this directory and at the command line type:

```
shell> ./mysql-monitor-agent start
```

To stop the agent, use the `stop` command:

```
shell> ./mysql-monitor-agent stop
```

If the agent cannot be stopped because the `pid` file that contains the agent's process ID cannot be found, you can use `kill` to send a `TERM` signal to the running process:

```
shell> kill -TERM PID
```

If you are running more than one agent on a specific machine, you must also specify the path to the `ini` file when you are stopping the agent. Executing `mysql-monitor-agent stop` without an `ini` file will only stop the agent associated with the default `ini` file.

To verify that the agent is running use the following command:

```
shell> ./mysql-monitor-agent status
```

The resulting message indicates whether the agent is running or not. If the agent is not running, use the following command to view the last ten entries in the agent log file:

```
shell> tail /Applications/mysql/enterprise/agent/log/mysql-monitor-agent.log
```

For further information on troubleshooting the agent see Section 2.3.7, “Troubleshooting the Agent”.

Installation creates the directory `/Applications/mysql/enterprise/agent` with the settings stored in the `mysql-monitor-agent.ini` file located directly below this directory in the `etc` directory. The `log` directory is also located immediately below the `agent` directory.

To see all the command-line options available when running the monitor agent, navigate to the `/Applications/mysql/enterprise/agent/etc/init.d` directory and execute **`mysql-monitor-agent`** `help`. You should see the message:

```
Usage: ./mysql-monitor-agent {start|stop|restart|status} [ini-file-name]
```

The `ini-file-name` option only needs to be used if the `ini` file is not installed to the default location or you have changed the name of the `ini` file. You will need to use this option if you are installing more than one agent on the same machine. Pass the full path to the `ini` file. For example, after navigating to the `/Applications/mysql/enterprise/agent/etc/init.d` directory, issue the command:

```
shell> ./mysql-monitor-agent start /Applications/mysql/enterprise/agent/etc/new-mysql-monitor-agent.ini
```

If you installed the agent as `root`, on reboot the `mysql-monitor-agent` daemon will start up automatically. If you installed the agent as an unprivileged user, you must manually start the agent on reboot or write a script to perform this task. Likewise, if you have added an additional agent as described in Section 2.3.6.2, “MySQL Server (`agent-instance.ini`) Configuration”, and you wish to start this agent on reboot, create a system initialization script appropriate to your operating system. To determine whether the agent is running or not navigate to the `init.d` directory and issue the command **`./mysql-monitor-agent status`**.

> **Warning**
>
> To report its findings, the agent needs to be able to connect to the dashboard through the port specified during installation. The default value for this port is `18080`; ensure that this port is not blocked. If you need help troubleshooting the agent installation see, Section 2.3.7, “Troubleshooting the Agent”.

2.3.5.3. Starting/Stopping the Agent on Unix

When installation is finished, you can start the monitor agent from the command line by typing:

```
shell> /opt/mysql/enterprise/agent/etc/init.d/mysql-monitor-agent start
```

For a non-`root` installation the command would be:

```
shell> /home/<user name>/mysql/enterprise/agent/etc/init.d/mysql-monitor-agent start
```

To stop the agent, use the `stop` command:

```
shell> ./mysql-monitor-agent stop
```

If the agent cannot be stopped because the `pid` file that contains the agent's process ID cannot be found, you can use `kill` to send a `TERM` signal to the running process:

```
shell> kill -TERM PID
```

If you are running more than one agent on a specific machine, you must also specify the path to the `ini` file when you are stopping the agent. Executing `mysql-monitor-agent stop` without an `ini` file will only stop the agent associated with the default `ini` file. Likewise, when checking the status of an agent specify its `ini` file.

To verify that the agent is running use the following command:

```
shell> ./mysql-monitor-agent status
```

The resulting message indicates whether the agent is running or not. If the agent is not running, use the following command to view the last ten entries in the agent log file:

```
shell> tail /opt/mysql/enterprise/agent/log/mysql-monitor-agent.log
```

For further information on troubleshooting the agent see Section 2.3.7, “Troubleshooting the Agent”.

Installation creates the directory `/opt/mysql/enterprise/agent` with the settings stored in the `mysql-monitor-agent.ini` file located directly below this directory in the `etc` directory. The `log` directory is also located immediately below the `agent` directory.

To see all the command-line options available when running the monitor agent, navigate to the /opt/mysql/enterprise/agent/etc/init.d directory and execute **mysql-monitor-agent** help. You should see the message:

```
Usage: ./mysql-monitor-agent {start|stop|restart|status} [ini-file-name]
```

The ini-file-name option only needs to be used if the ini file is not installed to the default location or you have changed the name of the ini file. You will need to use this option if you are installing more than one agent on the same machine. Pass the full path to the ini file. For example, after navigating to the /opt/mysql/enterprise/agent/etc/init.d directory, issue the command:

```
shell> ./mysql-monitor-agent start /opt/mysql/enterprise/agent/etc/new-mysql-monitor-agent.ini
```

If you installed the agent as root, on reboot the mysql-monitor-agent daemon will start up automatically. If you installed the agent as an unprivileged user, you must manually start the agent on reboot or write a script to perform this task. Likewise, if you have added an additional agent as described in Section 2.3.6.2, "MySQL Server (agent-instance.ini) Configuration", and you wish to start this agent on reboot, create a system initialization script appropriate to your operating system. To determine whether the agent is running or not navigate to the init.d directory and issue the command **./mysql-monitor-agent status**.

> **Warning**
>
> To report its findings, the agent needs to be able to connect to the dashboard through the port specified during installation. The default value for this port is 18080; ensure that this port is not blocked. If you need help troubleshooting the agent installation see, Section 2.3.7, "Troubleshooting the Agent".

2.3.6. Advanced Agent Configuration

The MySQL Enterprise Monitor Agent is configured through files located within the etc directory within the directory where you installed the agent.

Configuration is stored in multiple files, according to a predetermined file and directory layout. The primary configuration file contains specific information about the agent and how the agent communicates with MySQL Enterprise Service Manager. The main configuration is located within the mysql-monitor-agent.ini file.

Additional configuration files contain information about the MySQL server that is being monitored. You can configure which directory is used for storing this information within the mysql-monitor-agent.ini file. The default location is the etc/instances directory within the MySQL Enterprise Monitor Agent directory.

The server you want to monitor should have a directory within the specified location, optionally using the name of the server you are monitoring, and within that directory, an agent-instance.ini file. This file contains the configuration information for connecting to the MySQL server, including the host name, port, user credentials and display name.

You can see an example of the file layout of the etc directory:

```
.
./init.d
./init.d/mysql-monitor-agent
./instances
./instances/agent
./instances/agent/agent-instance.ini
./mysql-monitor-agent.ini
```

For more information on the configuration of the mysql-monitor-agent.ini file, see Section 2.3.6.1, "MySQL Enterprise Monitor Agent (mysql-monitor-agent.ini) Configuration". For details on the content of the individual MySQL instance configuration files, see Section 2.3.6.2, "MySQL Server (agent-instance.ini) Configuration".

2.3.6.1. MySQL Enterprise Monitor Agent (mysql-monitor-agent.ini) Configuration

The mysql-monitor-agent.ini files contains the base configuration information for the MySQL Enterprise Monitor Agent. The file sets the core information about the supported functionality for the entire agent.

You can see a sample of the configuration file below:

```
# WARNING - the UUID defined below must be unique for each agent.
#
# To use this .ini file as a template for configuring additional
# agents, do not simply copy and start a new agent without first
# modifying the UUID.
#
# Refer to the documentation for more detailed information and
# instructions.
#
# Version: 20080718_230416_r7011

[mysql-proxy]

plugins=proxy,agent
agent-mgmt-hostname = http://agent:password@monitor-server:18080/heartbeat
mysqld-instance-dir= etc/instances
agent-item-files = share/mysql-proxy/items/quan.lua,share/mysql-proxy/items/items-mysql-monitor.xml
proxy-address=:4040
proxy-backend-addresses = 127.0.0.1:3306
proxy-lua-script        = share/mysql-proxy/quan.lua


agent-uuid = 8770ead5-3632-4b29-a413-4a7c92437e26
log-file = mysql-monitor-agent.log
pid-file=/Applications/mysql/enterprise/agent/mysql-monitor-agent.pid
```

> **Note**
>
> Do not copy the agent configuration information from one machine to another without changing the `agent-uuid`. Each agent instance must have a unique agent id.

The main configuration information must be located within the `[mysql-proxy]` section of the configuration file. The main configurable parameters within this file are:

- `plugins`: Configures the plugins to be used by the agent. When monitoring servers you must have the `agent` plugin configured. If you want to support Query Analyzer then you must also have the `proxy` module enabled. Plugins should be specified as a comma separated list of plugin names.

  If you selected to support Query Analyzer during installation of the agent, the default value will be `proxy,agent`. If you disabled Query Analysis during installation, the default value will be `agent`.

- `log-level`: Sets the logging level of the agent. The default level is `critical`.

  Valid values for `log-level` are as follows:

  - `debug`: Provides detailed information about what the agent is doing and the information being provided by the agent to the MySQL Enterprise Service Manager.
  - `critical`: Lists critical messages highlighting problems with the agent.
  - `error`: Lists error messages.
  - `warning`: Provides only warning messages generated by the agent.
  - `message`: Provides information about the agent and basic processing information.
  - `info`: Provides messages used for informational purposes.

  > **Warning**
  >
  > Be careful when setting the `log-level` to `debug`. Doing this will rapidly increase the size of your `mysql-monitor-agent.log` file. To avoid disk space problems, put the log files on a different drive from your MySQL server and the MySQL Enterprise Dashboard.
  >
  > It is strongly recommended that you use a `log-level` of `critical` or `error` in a production server. Use the higher-levels to provide more detailed information only for debugging problems with your agent.
  >
  > Under Windows, if you restart the agent from the command line after setting the `log-level` to `debug`, extensive debug information is displayed to the console as well as to the log file.

- `agent-mgmt-hostname`: Sets the URL to use when reporting information. This value will be automatically set to your MySQL Enterprise Service Manager during installation.

- `mysqld-instance-dir`: Sets the directory where the configuration files that specify the MySQL servers to be monitored can be located.

- `agent-item-files`: Sets the information that is provided up to the MySQL Enterprise Service Manager when the agent is reporting status information. You should leave this item with the default setting of the `share/mysql-proxy/items/quan.lua` (which provides Query Analyzer data) and `share/mysql-proxy/items/items-mysql-monitor.xml` (which provides the core agent monitoring data).

- `proxy-address`: Sets the address, port number, or both for the proxy to listen to for connections. The setting is used when employing Query Analysis as the address/port that you must configure your application to use in place of your normal MySQL server. By default this item is set during installation.

  The default value is 4040. If you want to support a different local host name/IP address and port, specify the host name and the port number, separated by a colon.

- `proxy-backend-addresses`: Sets the host name and port number to be used when communicating the backend MySQL server when employing query analyzer. This is the MySQL server where packets from the client are sent when communicating with the proxy on the host name/port set by the `proxy-address`.

- `proxy-lua-script`: Sets the Lua script to be used by the proxy when forwarding queries. To use Query Analyzer, this parameter should be set to `share/mysql-proxy/quan.lua`. This is the default value.

- `agent-uuid`: Sets the UUID (Universally Unique ID) of the agent. This value should be unique for all agents communicating with the same server, as the UUID is used to uniquely ID the agent within MySQL Enterprise Service Manager

  If you are setting up multiple hosts and copying the configuration between hosts, make sure that the `agent-uuid` is unique. You can have the agent create a new UUID by leavig this configuration property blank.

- `log-file`: Sets the location of the log file used to record information about the agent when it is running. If you do not specify a full path name, then the log file location is considered to be relative to the installation directory of the agent.

- `pid-file`: Sets the location of the file used to record the Process ID of the agent. This is used by the script that shuts down the agent to identify the process to be shutdown. The default value is the `mysql-monitor-agent.pid` file within the base installation directory as created by the agent installer.

2.3.6.2. MySQL Server (`agent-instance.ini`) Configuration

For the MySQL server that you want to monitor, you must create an `agent-instance.ini` within the directory specified by the `mysqld-instance-dir` configuration parameter within the main `mysql-monitor-agent.ini` file.

The `agent-instance.ini` file contains the host name and user credentials for connecting to the MySQL server that you want the agent to monitor. The format of the file is as follows:

```
# To use this .ini file as a template for configuring additional
# instances to monitor, do not simply copy and start a new agent
# without first modifying the displayname.
#
# Refer to the documentation for more detailed information and
# instructions.
#
# Version: 20080718_230416_r7011

[mysqld]
hostname = 127.0.0.1
port     = 3306
user     = root
password =
```

The individual configuration parameters can be defined as follows:

- `hostname`: The host name of the MySQL server that you want to monitor.

- `port`: The TCP/IP port of the MySQL server that you want to monitor.

- `user`: The user to use when connecting to the MySQL server that you want to monitor.

- `password`: The corresponding password to use when connecting to the MySQL server that you want to monitor.

It is also possible to configure the agent to use sockets. This can be done during installation by selecting “socket” rather than “TCP/IP” from the menu and then specifying the socket name. This can also be configured after installation by editing the `agent-instance.ini` configuration file, and adding the line:

```
socket = /full/path/to/mysql.sock
```

2.3.6.3. Monitoring Multiple MySQL Servers

You can monitor multiple MySQL servers (either on the same machine, or across different machines) using two different methods:

- By using a single agent instance to monitor multiple MySQL servers. You can use this method if you want to monitor multiple servers, but do not want or need to support Query Analysis on the additional servers.
- By using multiple copies of the MySQL Enterprise Monitor Agent to monitor each server individually. Using this method requires additional overhead to monitor each server, while also allowing you to supply Query Analyzer data.

Using a Single Agent Instance

> **Warning**
>
> Do not use the single agent instance method if you want to use Query Analyzer. If you set your application to use the proxy port provided by the single instance then the queries may not be directed to the correct server. Using Query Analyzer, the proxy, and the single agent instance method is not supported.

> **Warning**
>
> When using the single agent instance method, the agent will attempt to determine the right information about the backend server that it is monitoring to use the information when applying rule and advisor information. Currently, this operation is performed for only one of the servers in the list of configured servers. If the servers being monitoring are using different MySQL versions then the rules applied to the servers may be incorrect, and you could get wrong or misleading advice about issues or problems on a given server.

To use a single agent to monitor multiple instances, you can create additional directories and configuration files within the `instances` directory for the agent. For example, you can see the default structure of the agent configuration directory:

```
./init.d
./init.d/mysql-monitor-agent
./instances
./instances/agent
./instances/agent/agent-instance.ini
./mysql-monitor-agent.ini
```

Within the `instances` directory, you can add further directories, one for each monitored server. Each additional directory must have a suitable `agent-instance.ini` file containing the connection information for the new MySQL server instance. For example, the following structure demonstrates an agent monitoring four MySQL servers:

```
./init.d
./init.d/mysql-monitor-agent
./instances
./instances/agent
./instances/agent/agent-instance.ini
./instances/mysql2
./instances/mysql2/agent-instance.ini
./instances/mysql-rep
./instances/mysql-rep/agent-instance.ini
./instances/mysql-backup
./instances/mysql-backup/agent-instance.ini
./mysql-monitor-agent.ini
```

To add another MySQL monitored server, follow these steps:

1. Make sure that the MySQL instance that you want to monitor has a suitable user to use for connecting to the server. For more information, see Section 2.3.1, “Creating a MySQL User Account for the Monitor Agent”.
2. Copy an existing configuration directory and configuration files to the new directory:

```
shell> cp -R etc/instances/agent etc/instances/mysql2
```

3. Edit the configuration file within the new directory, for example `mysql2/agent-instance.ini`, and set the `user`, `password` and either the `hostname` and `port`, or `socket` parameters.

4. Restart the agent:

   ```
   shell> mysql-monitor-agent restart
   ```

Using Multiple Agent Instances

To use multiple agents to monitor multiple MySQL servers you need to create a new configuration structure for both the agent and the MySQL server instances you need to monitor, including the binaries and configuration files, and then update the configuration to set the corresponding parameters to monitor the new server. Using this method allows you to enable query analyis by redirecting requests to the target server using the built-in proxy service within the agent.

For example, the directory structure below shows the configuration directory for two agents monitoring a single MySQL server each:

```
./init.d
./init.d/mysql-monitor-agent
./instances
./instances/agent
./instances/agent/agent-instance.ini
./instances-second/agent
./instances-second/agent/agent-instance.ini
./mysql-monitor-agent.ini
./mysql-second-agent.ini
```

The `mysql-monitor-agent.ini` file contains the configuration for the first agent, with the MySQL servers monitored defined within the `instances` directory. The `mysql-second-agent.ini` file contains the configuration information for the second agent, with the MySQL servers monitor defined within the `instances-second` directory.

To set up multiple agents:

1. Make sure that the MySQL instance that you want to monitor has a suitable user to use for connecting to the server. For more information, see Section 2.3.1, “Creating a MySQL User Account for the Monitor Agent”.

2. You need to generate a new UUID for the new agent:

   ```
   shell> /opt/mysql/enterprise/agent/bin/mysql-monitor-agent --agent-generate-uuid
   ee9296d7-f7cd-4fee-8b26-ead884ebf398
   2009-03-05 11:49:37: (critical) shutting down normally
   ```

   Keep a record of the UUID to update the configuration file.

   Note, the agent should not be running when the UUID is generated.

3. Copy the main agent configuration file, which is by default in `/opt/mysql/enterprise/agent/etc/mysql-monitor-agent.ini`:

   ```
   shell> cp mysql-monitor-agent.ini mysql-second-agent.ini
   ```

4. Edit the new configuration file, changing the following settings:

   - Change the `mysqld-instance-dir` to the new directory that will contain the individual MySQL server configuration files.
   - Change the `proxy-address` to a different value than the first agent configuration.
   - Change the `proxy-backend-addresses` to specify the IP address and MySQL port number for the MySQL server.
   - Change the `agent-uuid` to the new value obtained in an earlier step.
   - Change the `log-file` parameter to specify a different file to use when logging errors and problems. You cannot log to the same file from two different agents.

- Change the pid-file parameter to specify the file that will be used to store the process ID of the agent.

5. Copy an existing configuration directory and configuration files to the new directory:

   ```
   shell> cp -R etc/instances etc/instances-second
   ```

6. Edit the configuration file, instances/second/agent/agent-instance.ini within the new directory, and set the user, password and either the hostname and port, or socket parameters.

7. With multiple instances, you must start each agent individually, specifying the location of the main configuration file. For example, to start the original (default) service:

   ```
   shell> /opt/mysql/enterprise/agent/etc/init.d/mysql-monitor-agent start /opt/mysql/monitor/agent/etc/mysql-monitor-age
   ```

   To start the second instance:

   ```
   shell> /opt/mysql/enterprise/agent/etc/init.d/mysql-monitor-agent start /opt/mysql/monitor/agent/etc/mysql-second-agen
   ```

2.3.6.4. Configuring an Agent to Monitor a Remote MySQL Server

Typically, the agent runs on the same machine as the MySQL server it is monitoring. Fortunately, this is not a requirement. If you want to monitor a MySQL server running on an operating system for which there is no agent available, you can install the agent on a machine other than the one hosting the MySQL server.

The process for installing an agent to monitor a MySQL server on a remote machine is identical to the process described in Section 2.3, “Monitor Agent Installation”. Follow the directions given there, being careful to specify the correct IP address or host name for the MySQL Enterprise Service Manager and likewise for the MySQL server—since the agent is not running on the same machine as the MySQL server, it cannot be the default, localhost.

Don't forget that the agent must be given rights to log in to the MySQL server from a host other than localhost and that the port used by the MySQL server, typically 3306 must be open for remote access. For more information about the database credentials required by agents see, Section 2.3.1, “Creating a MySQL User Account for the Monitor Agent”.

The agent also needs to be able to log in to the MySQL Enterprise Service Manager, typically using port 18080, so ensure that the appropriate port is open.

> **Note**
>
> Remote agents do not report the OS information for either the host or the agent.
>
> If your subscription level entitles you to replication autodiscovery, do **not** use remote monitoring with replication slaves or masters. The agent must be installed on the same machine as the server you are monitoring for discovery to work properly. For more information, see Chapter 9, *The Replication Page*.

2.3.6.5. Monitoring Outside the Firewall with an SSH Tunnel

If you run an SSH server on the machine that hosts the MySQL Enterprise Service Manager and an SSH client on the machine that hosts the agent, you can create an SSH tunnel so that the agent can bypass your firewall. First, you need to make an adjustment to the hostname value specified in the [mysql-proxy] section of the .ini file. (For more information about the contents and location of the .ini file see Section 2.3.6.1, “MySQL Enterprise Monitor Agent (mysql-monitor-agent.ini) Configuration”.) Stop the agent and change the hostname value as shown in the following:

```
hostname = http://agent_name:password@localhost:18080/heartbeat
```

Replace the agent_name and password with suitable values. Likewise replace port 18080 if you are not running the dashboard on this port. Use localhost for the host name, since the agent is connecting through an SSH tunnel.

Next, execute the following command on the machine where the agent is running:

```
shell> ssh -L 18080:Dashboard_Host:18080 -l user_name -N Dashboard_Host
```

When prompted, enter the password for *`user_name`*.

If you are not running the MySQL Enterprise Service Manager on port `18080`, substitute the appropriate port number. Likewise, replace `Dashboard_Host` with the correct value. `user_name` represents a valid operating system user on the machine that hosts the MySQL Enterprise Service Manager.

Be sure to restart the agent so that the new value for the `hostname` takes effect. For instructions on restarting the agent see:

- Under Windows see, Section 2.3.5.1, “Starting/Stopping the Agent on Windows”.
- Under Unix see, Section 2.3.5.3, “Starting/Stopping the Agent on Unix”.
- Under Mac OS X see, Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”.

2.3.6.6. Generating a new UUID

For MySQL Enterprise Monitor to operate correctly, each agent must have a unique UUID to uniquely identify the agent with the MySQL Enterprise Service Manager.

> **Warning**
>
> Ensure that you do not reuse or duplicate a UUID. Running two agents with the same identification number yields unpredictable results

In Unix go to the command line and type:

```
shell> /opt/mysql/enterprise/agent/bin/mysql-monitor-agent --agent-generate-uuid
```

In Mac OS X go to the command line and type:

```
shell> /Applications/mysql/enterprise/agent/bin/mysql-monitor-agent --agent-generate-uuid
```

This should display a line similar to the following:

```
ee9296d7-f7cd-4fee-8b26-ead884ebf398
```

Paste this line into the `[mysql-proxy]` section of the `mysql-monitor-agent.ini` file for the `agent-uuid` parameter:

```
[mysql-proxy]
...
agent-uuid=ee9296d7-f7cd-4fee-8b26-ead884ebf398
```

In Windows, go to the command line and change to the MySQL Enterprise Monitor Agent installation directory and update the UUID by executing `mysql-monitor-agent -uf mysql-monitor-agent-3307.ini`. For example:

```
C:\> cd C:\Program Files\MySQL\Enterprise\Agent
C:\> mysql-monitor-agent -uf mysql-monitor-agent.ini (or your .ini file name)
```

This updates the configuration file directly with the new UUID.

2.3.7. Troubleshooting the Agent

The first step in troubleshooting the agent is finding out whether it is running or not. To do this see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

If incorrect credentials are specified for the agent login to the MySQL server that it is monitoring, then the agent will not run on startup. Log in to the monitored MySQL server and check the agent's credentials. Compare the values of the Host, User, and Password fields in the mysql.user table with the values shown in the [mysqld] section of the etc/instances/mysql/agent-instance.ini. If incorrect credentials are specified in the ini file, simply correct them and restart the agent. Remember, changes to the ini file do not take effect until the agent is restarted.

The agent will not start up if incorrect credentials are specified for the service manager login. Using incorrect credentials for logging in to the service manager creates an entry in the agent log file. For the location of this log file see Section C.3, “Agent Log and PID Files”.

If the agent starts up but no server appears in the dashboard, check the hostname specified in the [mysql-proxy] portion of the mysql-monitor-agent.ini file. Incorrect credentials, IP address, or port will all cause the MySQL server to fail to appear in the dashboard. Also, ensure that the port specified in this file is not blocked on the machine hosting the MySQL Enterprise Service Manager.

An easy way to confirm that the agent can log in to the service manager is to type ***http://Dashboard_Host:18080/*** **heartbeat** into the address bar of your web browser, substituting the appropriate host name and port. When the HTTP authentication dialog box opens, enter the agent user name and password. If you log in successfully, you should see the following message:

```
<exceptions>
<error>E1031:  Agent payload parameter NULL.</error>
</exceptions>
```

Note

Despite the fact that the preceding listing shows an error, you have logged in successfully. This error appears *because* you have logged in but with no “payload”.

If you can log in successfully in the way described above and the agent is running, then there are errors in the mysql-monitor-agent.ini file. Compare the host name, port, agent name, and password found in the ini file with the values you entered into the address bar of your web browser.

If HTTP authentication fails then you are using incorrect credentials for the agent. Attempting to log in to the service manager using incorrect credentials creates an entry in the agent log file. For the location of this log file see Section C.3, “Agent Log and PID Files”.

If no HTTP authentication dialog box appears, and you are unable to connect at all, then you may have specified an incorrect host name or port. Confirm the values you entered against those described as the Application hostname and port: in the configuration_report.txt file. Failure to connect could also indicate that the port is blocked on the machine hosting the MySQL Enterprise Service Manager.

To check if a blocked port is the problem, temporarily bring down your firewall. If the agent is then able to connect, open up the port specified during installation and restart the agent. If necessary you can monitor outside the firewall using an SSH tunnel. For more information, see Section 2.3.6.5, “Monitoring Outside the Firewall with an SSH Tunnel”.

You can also check the agent error log file to help determine any problems. An error such as the following might indicate a blocked port:

```
(critical) connection to  merlin-server
'http://agent:test@172.11.1.1:18080/heartbeat' failed:
"connect() timed out!" error.
```

For the location of the agent error log file see, Section C.3, “Agent Log and PID Files”.

Setting the log-level entry in your ini file is also a good debugging technique. For more information on this subject see, Section 2.3.6.1, “MySQL Enterprise Monitor Agent (mysql-monitor-agent.ini) Configuration”.

Running the agent from the command line sometimes displays errors that fail to appear in the log file or on the screen when the agent is started from a menu option. To start the agent from the command line see the instructions given at the start of this section.

If you have more than one agent running on the same machine, the UUID must be unique and the log-file and pid-file values must be different. For more information, see Section 2.3.6.2, “MySQL Server (agent-instance.ini) Configuration”.

If the agent is not running on the same machine that hosts the MySQL server it is monitoring, then you must ensure that the correct host is specified for the agent account. The correct port, typically 3306, must also be open for remote login. For more information about remote monitoring see, Section 2.3.6.4, “Configuring an Agent to Monitor a Remote MySQL Server”.

If the MySQL Enterprise Monitor Agent has been ungracefully terminated and restarted (for example after being terminated using kill), then you may see a DuplicateAgentUuidException error until the original registration of the previous instance of the agent has expired.

2.4. Unattended Installation

It is possible to install the MySQL Enterprise Monitor without any direct user interaction. This is done by passing the command-line option --mode unattended to the installation file.

Using this mode and other command-line parameters means the user will not be prompted for input during installation. This is especially useful when doing multiple installations of the MySQL Enterprise Monitor.

However, rather than passing numerous parameters from the command line, it is usually more convenient to save your options in a text file and invoke the installer using the optionfile option. This is a more reusable and less error-prone solution.

Before attempting an unattended installation, it is recommended that you install the MySQL Enterprise Monitor interactively at least once. Failing this, as a minimum, read the regular installation instructions since some tasks still remain after an unattended installation; you must configure the MySQL Enterprise settings, import the advisors, and start up all the services/daemons.

2.4.1. Command-Line Options

To view the available options for the monitor installer or for the agent installer, at the command line type the executable file name along with the --help option.

On each platform, for each installer, the installer supports a number of different installation modes. Some of these are unique to an individual platform, others are available on all platforms. The table below summarizes the different options available for each platform.

Platform	**Default Mode**	**win32**	**gtk**	**xwindow**	**osx**	**text**	**unattended**
Windows	win32	Y	N	N	N	N	Y
Unix	gtk	N	Y	Y	N	Y	Y
Linux	gtk	N	Y	Y	N	Y	Y
Mac OS X	win32	N	N	N	Y	Y	Y

2.4.1.1. MySQL Enterprise Service Manager Options

The following listing shows the command line options for the MySQL Enterprise Service Manager.

```
--help                         Display the list of valid options

--version                      Display product information

--optionfile <optionfile>      Installation option file

                               Default:

--mode <mode>                  Installation mode
                               Default: win32
                               Allowed: win32 unattended

--debugtrace <debugtrace>      Debug filename
                               Default:

--installer-language <installer-language> Language selection
                               Default:
                               Allowed: en  jp

--installdir <installdir>      Installation directory
                               Default:/opt/mysql/enterprise/monitor/

--tomcatport <tomcatport>      Tomcat Server Port
                               Default: 18080

--tomcatshutdownport <tomcatshutdownport> Tomcat Shutdown Port
                               Default: 18005

--tomcatsslport <tomcatsslport>Tomcat SSL Port
                               Default: 18443

--usessl <usessl>              Should communication between the Dashboard »
                                      and Service Manager be encrypted?
                               Default: 0
```

```
--adminuser <adminuser>        Repository Username
                               Default: service_manager


--adminpassword <adminpassword>Password
                               Default:

--dbport <dbport>              Bundled MySQL Database Port
                               Default: 13306
```

The options and their effect on installation are detailed below:

- `--help`

  Display the list of valid options.

- `--version`

  Display product and version information.

- `--optionfile`

  The path to the option file containing the information for the installation.

- `--mode`

  The installation mode to be used for this installation.

- `--debugtrace`

  The filename to be used for a debug trace of the installation.

- `--installer-language`

  The installer language; supported options are `en` for English and `jp` Japanese.

- `--installdir`

  The installation directory for MySQL Enterprise Service Manager.

  The default on Windows is `C:\Program Files\MySQL\Enterprise\Monitor`

  The default on Unix is `/opt/mysql/enterprise/monitor/`

  The default on Mac OS X is `/Applications/mysql/enterprise/monitor/`

- `--tomcatport`

  The MySQL Enterprise Service Manager port;

  The default is 18080.

- `--tomcatshutdownport`

  The MySQL Enterprise Service Manager Tomcat shutdown port.

  The default is 18005.

- `--tomcatsslport`

  The MySQL Enterprise Service Manager SSL port.

  The default is 18443.

- `--usessl`

Enable support for SSL communication between the MySQL Enterprise Monitor Agent and MySQL Enterprise Service Manager.

The default is 0.

- `--adminuser`

  The MySQL Enterprise Service Manager user name.

  The default is `service_manager`.

  > **Warning**
  >
  > The repository user name and password are stored in unencrypted form in the `config.properties` file. To locate this file on your operating system see Section C.5, "The `config.properties` File".

- `--adminpassword`

  The MySQL Enterprise Service Manager password.

- `--dbport`

  The TCP/IP port for the Bundled MySQL database.

  The default is 13306.

2.4.1.2. MySQL Enterprise Monitor Agent Options

To view all the options available for an unattended *agent* installation, invoke the agent installer file passing in the `help` option. (Under Windows you must redirect the output to a file. You should see a listing similar to the following:

> **Note**
>
> The exact options may vary depending on the operating system on which you are executing the installer.

The options and their effect on installation are detailed below:

- `--help`

  Display the list of valid options

- `--version`

  Display product information, including the version number of the installer.

- `--optionfile <optionfile>`

  Specify the location of an option file containing the configuration options for this installation.

- `--unattendedmodeui <unattendedmodeui>`

  The UI elements to use when performing an unattended installation. The options are `none`, show now UI elements during the installation; `minimal`, show minimal elements during installation; `minimalWithDialogs`, show minimal UI elements, but include the filled-dialog boxes.

  The default is `none`.

- `--mode <mode>`

  Specify the installation mode to use for this installation.

- `--debugtrace <debugtrace>`

  Set the filename to use when recording debug information during the installation.

- `--installer-language <installer-language>`

  Set the language to be used for the installer.

- `--installdir <installdir>`

  Specify the directory where the software will be installed.

  The default on Windows is `C:\Program Files\MySQL\Enterprise\Agent`

  The default on Unix is `/opt/mysql/enterprise/agent/`

  The default on Mac OS X is `/Applications/mysql/enterprise/agent/`

- `--mysqlconnmethod <mysqlconnmethod>`

  Specify the connection method to use to connect to MySQL.

  Options are `tcpip` and `socket`.

  The default is `tcpip`.

- `--mysqlhost <mysqlhost>`

  MySQL hostname or IP address

  The default is 127.0.0.1.

- `--checkmysqlhost <checkmysqlhost>`

  Validate the MySQL hostname or IP address

  The default is `yes`.

- `--mysqlport <mysqlport>`

  Specify the TCP/IP port to use when connecting to MySQL.

  The default is 3306.

- `--mysqlsocket <mysqlsocket>`

  Specify the filename of the MySQL socket to use when communicating with the monitored MySQL instance.

- `--mysqluser <mysqluser>`

  Specify the username to use when connecting to the MySQL instance.

- `--mysqlpassword <mysqlpassword>`

  Specify the password to use when connecting to the MySQL instance.

- `--enableproxy <enableproxy>`

  Enable the Proxy. This is recommended and is required if you want to use Query Analyzer.

  The default is 1 (use the proxy).

- `--proxyport <proxyport>`

  Specify the TCP/IP port to use for the proxy interface.

  The default is 4040.

- `--managerhost <managerhost>`

  The hostname or IP address of the MySQL Enterprise Service Manager.

- `--managerport <managerport>`

  The port number of the MySQL Enterprise Service Manager.

  The default is 18080.

- `--managersslport <managersslport>`

  The port number of the MySQL Enterprise Service Manager for SSL-based communication

  The default is 18443.

- `--usessl <usessl>`

  Specifies whether SSL should be used to communicate with the MySQL Enterprise Service Manager.

- `--agentuser <agentuser>`

  Specify the agent username to be used when communicating with the MySQL Enterprise Service Manager.

- `--agentpassword <agentpassword>`

  Specify the agent password to be used when communicating with the MySQL Enterprise Service Manager.

- `--proxyuser <proxyuser>`

  The user account for the proxy server.

  The default is `root`.

2.4.2. Unattended Windows Installation

For unattended installation on Windows, create an option file named `options.server.txt`. The following is an example of what the contents of an option file might be.

```
debugtrace=C:\Program Files\MySQL\Enterprise\install.debugtrace.log
mode=unattended
installdir=C:\Program Files\MySQL\Enterprise
tomcatport=8080
tomcatshutdownport=8005
tomcatsslport=8443
adminpassword=myadminpassword
dbport=3300
```

This file identifies a directory and file name for a log file, sets the `mode` to `unattended`, and uses the `installdir` option to specify an installation directory. The meaning of the other options is fairly self-evident.

> **Note**
>
> Set the `installdir` and `debugtrace` options to values appropriate to your locale and operating system.
>
> The only options that must be specified in an option file when installing the MySQL Enterprise Service Manager are `mode` (if not specified at the command line), `installdir`, and `adminpassword`.
>
> Check the options in your option file closely before installation; no warnings will be issued if there are errors.

Ensure that the monitor installer file and the options file are in the same directory and, if you saved the options file as `options.server.txt`, you can invoke an unattended installation from the command line by typing:

```
  C:\ mysqlmonitor-version-windows-installer.exe --optionfile options.server.txt
```

You can install the MySQL Enterprise Monitor Agent in exactly the same fashion. Create an agent option file and call the agent installer using the `optionfile` option.

As a minimum for the agent installation, you must specify the `mode` (if not specified at the command line), `mysqluser`, `installdir`, `mysqlpassword`, `installdir`, `managerhost`, and `agentpassword` options. Create a file containing these values and use it with the optionfile option for unattended agent installation.

If you wish, you can create one script that calls both the Service Manager and the Monitor Agent programs passing appropriate optionfile options.

2.4.3. Unattended Unix and Mac OS X Installation

For unattended installation on Unix, create an option file named options.server.txt. The following is an example of what the contents of an option file might be for installation on Unix.

```
debugtrace=/opt/mysql/enterprise/install.debugtrace.monitor.log
mode=unattended
installdir=/opt/mysql/enterprise/monitor
tomcatport=8080
tomcatshutdownport=8005
tomcatsslport=8443
adminpassword=myadminpassword
dbport=3300
```

This file identifies a directory and file name for a log file, sets the mode to unattended, and uses the installdir option to specify an installation directory. The meaning of the other options is fairly self-evident.

> **Note**
>
> Set the installdir and debugtrace options to values appropriate to your locale and operating system.
>
> The only options that must be specified in an option file when installing the MySQL Enterprise Service Manager are mode (if not specified at the command line), installdir, and adminpassword.
>
> Check the options in your option file closely before installation; no warnings will be issued if there are errors.

Ensure that the monitor installer file and the options file are in the same directory and, if you saved the options file as options.server.txt, you can invoke an unattended installation from the command line by typing:

```
  shell> mysqlmonitor-version-installer.bin --optionfile options.server.txt
```

You can install the MySQL Enterprise Monitor Agent in exactly the same fashion. Create an agent option file and call the agent installer using the optionfile option.

As a minimum for the agent installation, you must specify the mode (if not specified at the command line), mysqluser, installdir, mysqlpassword, and agentpassword options. Create a file containing these values and use it with the optionfile option for unattended agent installation.

If you wish, you can create one script that calls both the Service Manager and the Monitor Agent programs passing appropriate optionfile options.

> **Note**
>
> The Service Manager does not automatically start up on rebooting. For more information, see Bug#31676.

The procedure for unattended agent installation under Mac OS X is identical to the procedure under Unix.

2.4.4. Starting the Services

For instructions on starting the services needed by the MySQL Enterprise Service Manager see, Section 2.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows” for Windows and, Section 2.2.6, “Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X” for Unix and Mac OS X.

For instructions on starting the MySQL Enterprise Monitor Agent see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

If you wish, you can script the startup of these services.

2.5. Post-Installation Considerations

Depending upon how you plan to use the MySQL Enterprise Monitor, there are some tasks you may want to perform after installation. Find some suggestions in the following list:

- **Email settings**: Test email notification by deliberately triggering an alert.
- **Auto Startup**: On Unix systems, the MySQL Enterprise Service Manager does not automatically restart when the system is rebooted. You may wish to create a system initialization script appropriate to your operating system.
- **Log files**: Check the log files for any irregularities. For the locations of the various log files see Appendix C, *Files Associated with The MySQL Enterprise Monitor*.
- **Agent Log file rotation**: Implement log file rotation for the monitor agent.
- **Back up the repository**: For a back-up strategy suitable to your circumstances, see the MySQL reference manual documentation.
- **Configuration backup**: Back up the `mysql-monitor-agent.ini` file and the associated `instances` directory and contents.

  For more information about the `mysql-monitor-agent.ini` file see Section 2.3.6, “Advanced Agent Configuration”.
- **Configuration file**: Store the `configuration_report.txt` in a safe place. There is no mechanism for retrieving the password stored in this file.
- **Repository credentials**: The repository user name and password are stored in unencrypted form in the `config.properties` file. Take care to protect this file.
- **Disk management**: Remove installation files, and monitor the space used by the repository. Ensure that you have adequate disk space by regularly purging data. For more information, see Data Purge Behavior.
- **Firewall changes**: You may want to limit or expand access to the MySQL Enterprise Service Manager.
- **Open ports**: As with firewall changes, you may want to limit or expand access to the MySQL Enterprise Service Manager. The dashboard uses nonstandard ports, none of which are usually open by default.
- **Server upgrades**: See Section 2.6.3.1, “Upgrading the Monitored MySQL Server” for instructions on upgrading a server.
- **Repository access**: You may want to add other users.

2.6. Upgrading, Re-Installing or Changing Your Installation

You can upgrade MySQL Enterprise Monitor in a number of different ways:

- For instructions on upgrading your existing installation, see Section 2.6.1, “Upgrading MySQL Enterprise Monitor”.
- For more information on re-installing an existing installation, see Section 2.6.2, “Reinstalling MySQL Enterprise Monitor”.
- To change an existing installation, such as changing the monitored server, see Section 2.6.3, “Changing Your MySQL Enterprise Monitor Installation”.

2.6.1. Upgrading MySQL Enterprise Monitor

From time to time there may be updates to the MySQL Enterprise Service Manager or the MySQL Enterprise Monitor Agent. This section describes how to perform an update for either of these components.

You cannot use the update installers to change to a different operating system or chip architecture. For example, you cannot update a 32-bit Linux installation to a 64-bit version using an update installer—in cases such as this you must do a fresh installation.

The installation and configuration of MySQL Enterprise Monitor Agent must be standard before you start the installation. The update installer will not upgrade agents where you have changed or modified the filenames or directory layout of the installed agent, configuration files, or the startup files.

The name of the update file varies but it shows the target operating system and the version the update applies to. If a specific component is being updated it may also appear in the file name. For example, a file named *mysqlenterprisemanager-2.0.0-windows-update-installer.exe* would indicate a Windows update to MySQL Enterprise Service Manager version 2.0.0.

You may install an update in the same way that you initially installed the service manager or the agent; in win32 or unattended mode on Windows in gtk, text, xwindow, or unattended mode on Unix and in osx, text , or unattended mode on OS X.

> **Warning**
>
> The method you use for upgrading MySQL Enterprise Monitor components will depend on the upgrade you are performing.
>
> - If you are upgrading between major versions (for example, from MySQL Enterprise Monitor 1.3 to merlin 2.0), you should shutdown the MySQL Enterprise Service Manager and each connected MySQL Enterprise Monitor Agent. Once you have shutdown each component, start by updating the MySQL Enterprise Service Manager, and then updating the MySQL Enterprise Monitor Agent on each monitored client.
> - If you are upgrading between the same major version, for example, MySQL Enterprise Monitor 2.0 to MySQL Enterprise Monitor 2.1, or a minor version, such as MySQL Enterprise Monitor 2.1.1 to MySQL Enterprise Monitor 2.1.2, you can shutdown only the component (agent, or server) you are updating. Using this method, you can perform a 'rolling' upgrade, where you shutdown a single MySQL Enterprise Monitor Agent, upgrade it to the latest agent version, and then restart the agent before moving on to the next monitored instance.

> **Important**
>
> The upgrade installer will overwrite items-mysql-monitor.xml. On Windows this file is found in the C:\Program Files\MySQL\Enterprise\Agent\share\mysql-monitor-agent directory and on Unix in the /opt/mysql/enterprise/agent/share/mysql-monitor-agent directory. You should back this file up if you have made any changes to it.

> **Warning**
>
> If you use the Upgrade installer to update MySQL Enterprise Service Manager and you have made any changes to the my.cnf within your MySQL Enterprise Service Manager installation, any changes will be lost. You should copy the existing my.cnf file before starting the upgrade installer.

Otherwise, updating is a fairly straightforward process. Run the installation file and choose the directory of your current installation and whether or not you wish to back up your current installation. The time required to complete the process varies depending upon the nature of the update.

If you chose to back up your current installation, a directory named backup will be created in the current installation directory. This directory will contain copies of the directory or directories that were replaced during the update. In cases where only specific files are replaced, the backup directory may contain only these files. If you are unhappy with the update simply overwrite the new files or directories with the originals found in the backup directory. Be sure to stop both the MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent before restoring the original files. You can delete or archive this directory when you are satisfied that the update was successful.

If you choose to back up your current installation, the installer checks that there is adequate disk space for your repository backup. If there is not enough space, you are given the option of choosing another location; you may also choose not to back up the repository.

To update your Advisors see, Section 2.2.7.4, “Upgrading and Updating Advisors”.

2.6.1.1. Upgrading from MySQL Enterprise Monitor 1.3 to 2.0

To upgrade your existing installation from MySQL Enterprise Monitor 1.3 to MySQL Enterprise Monitor 2.0, you need to upgrade both your MySQL Enterprise Service Manager and your MySQL Enterprise Monitor Agent on each machine that you are monitoring.

To perform the update process you must use an update installer. This ensures that your current configuration information is migrated to the new version of MySQL Enterprise Service Manager.

Before you start the migration, shutdown your MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent on each monitored host. Then install the updated MySQL Enterprise Service Manager application to migrate the configuration and data of the main application and repository. Once the new MySQL Enterprise Service Manager is running, you can start to update and migrate each agent.

For more information on upgrading your MySQL Enterprise Service Manager, see Section 2.6.1.1.1, “Upgrading to MySQL Enterprise Service Manager 2.0”. For more information on upgrading an MySQL Enterprise Monitor Agent, see Section 2.6.1.1.2, “Upgrading to MySQL Enterprise Monitor Agent 2.0”.

2.6.1.1.1. Upgrading to MySQL Enterprise Service Manager 2.0

Upgrading MySQL Enterprise Service Manager requires you to use on of the *update* installers. The update installer performs a number of operations during installation:

- A new database, required to support 2.0 functionality, is created.
- You core dashboard, user, and rule information is migrated from the old database to the new database.
- The core configuration parameters for the MySQL Enterprise Service Manager are migrated from MySQL Enterprise Monitor 1.3 are migrated to MySQL Enterprise Monitor 2.0.

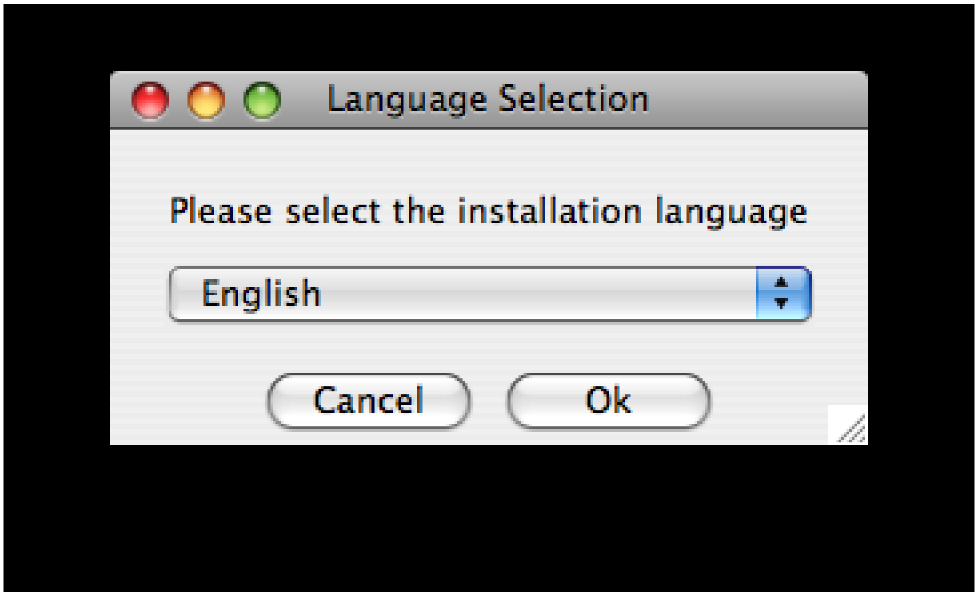
The installation of the new software using the update installer follows this basic sequence:

1. Request the installation language.
2. Confirm the location of the current MySQL Enterprise Service Manager installation.
3. Specify whether you want to keep a copy of the old server, application, and database files.
4. Configure the Tomcat server settings, including whether the new server should support SSL connections from agents.
5. If requested, the application and database information is backed up and upgraded, before the new application is installed.

The installation process is consistent for all platforms. A sample of the process for Max OS X has been provided below:

1. Double-click the update installer. The update installer will have `update` in the file name. For example, `mysqlmonitor-2.0.0.7101-osx-update-installer.app`.
2. Confirm the language you want to use when installing the software.

Figure 2.23. MySQL Enterprise Monitor: Server Update: Language Selection



Language Selection

Please select the installation language

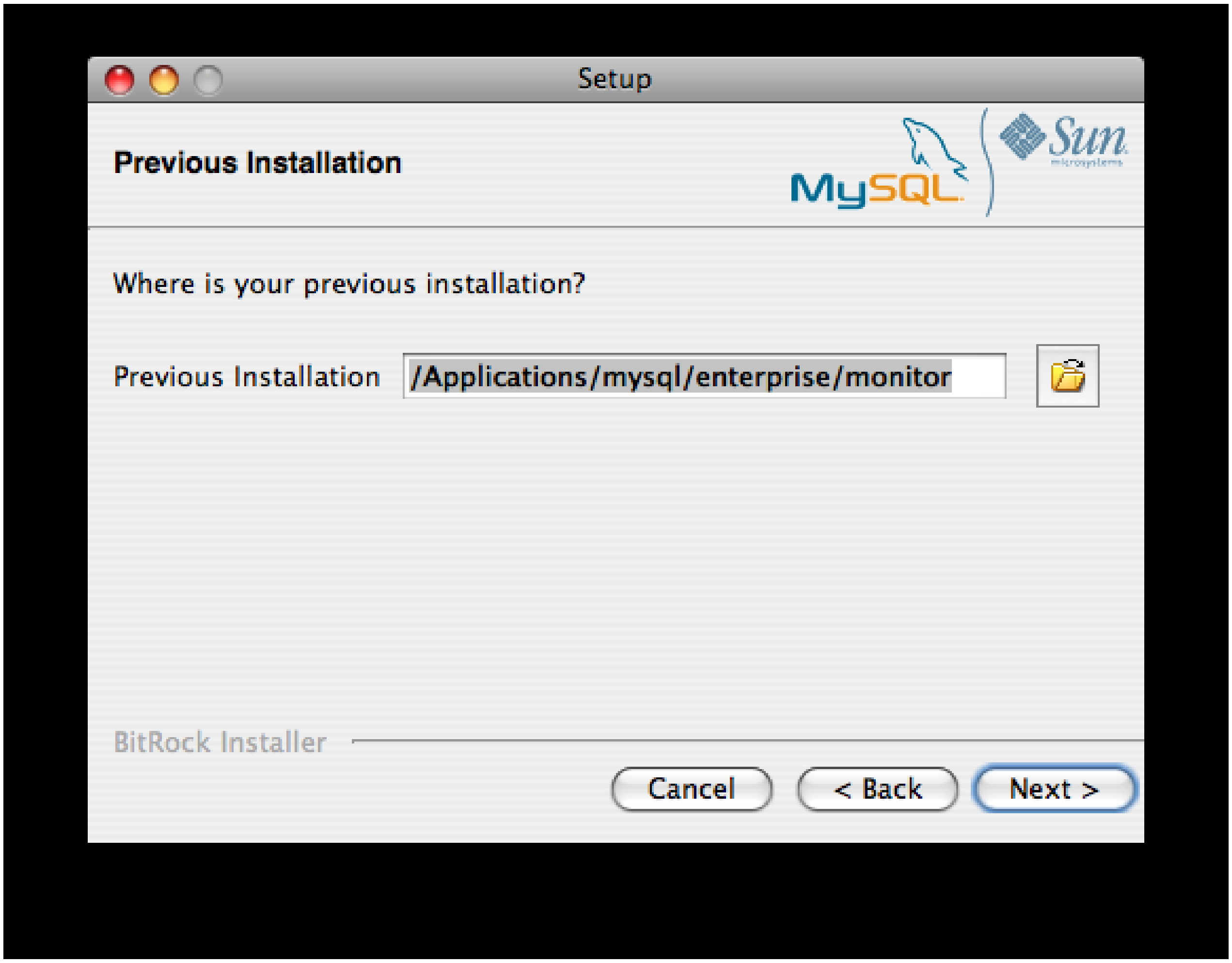
English

Click OK

3. You will be presented with an information screen showing the application you are installing. Click NEXT to continue.

4. Specify, or locate, the previous installation of MySQL Enterprise Service Manager If you installed the server within the default location, the current version of the application should be located automatically.

Figure 2.24. MySQL Enterprise Monitor: Server Update: Previous Installation



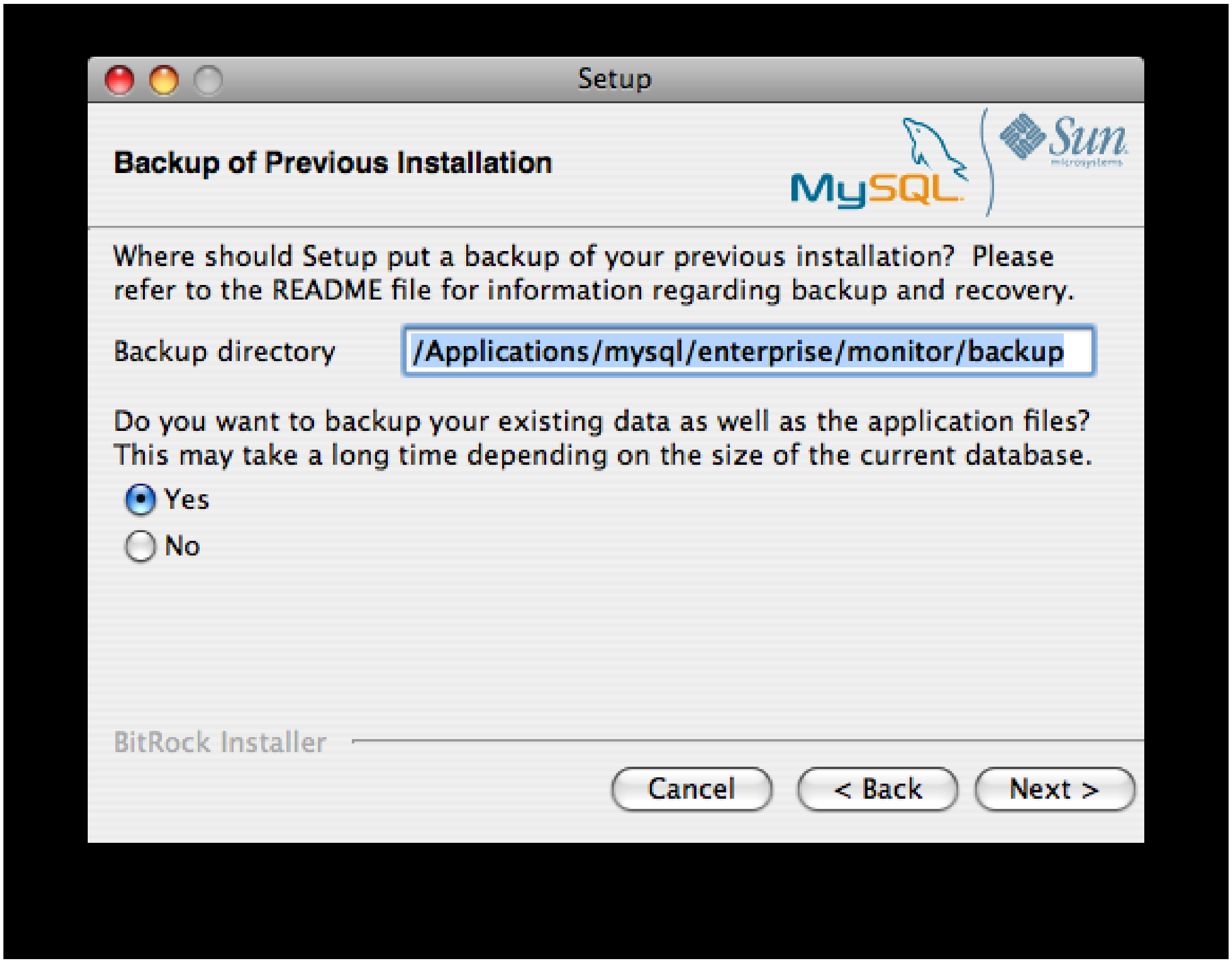
Previous Installation

Where is your previous installation?

Previous Installation /Applications/mysql/enterprise/monitor

5. The installer can keep a backup copy of your existing application, including keeping a complete backup of the data stored within your MySQL Enterprise Monitor repository database.

Figure 2.25. MySQL Enterprise Monitor: Server Update: Backup of Previous Installation



Setup

Backup of Previous Installation

Where should Setup put a backup of your previous installation? Please refer to the README file for information regarding backup and recovery.

Backup directory /Applications/mysql/enterprise/monitor/backup

Do you want to backup your existing data as well as the application files? This may take a long time depending on the size of the current database.

● Yes

○ No

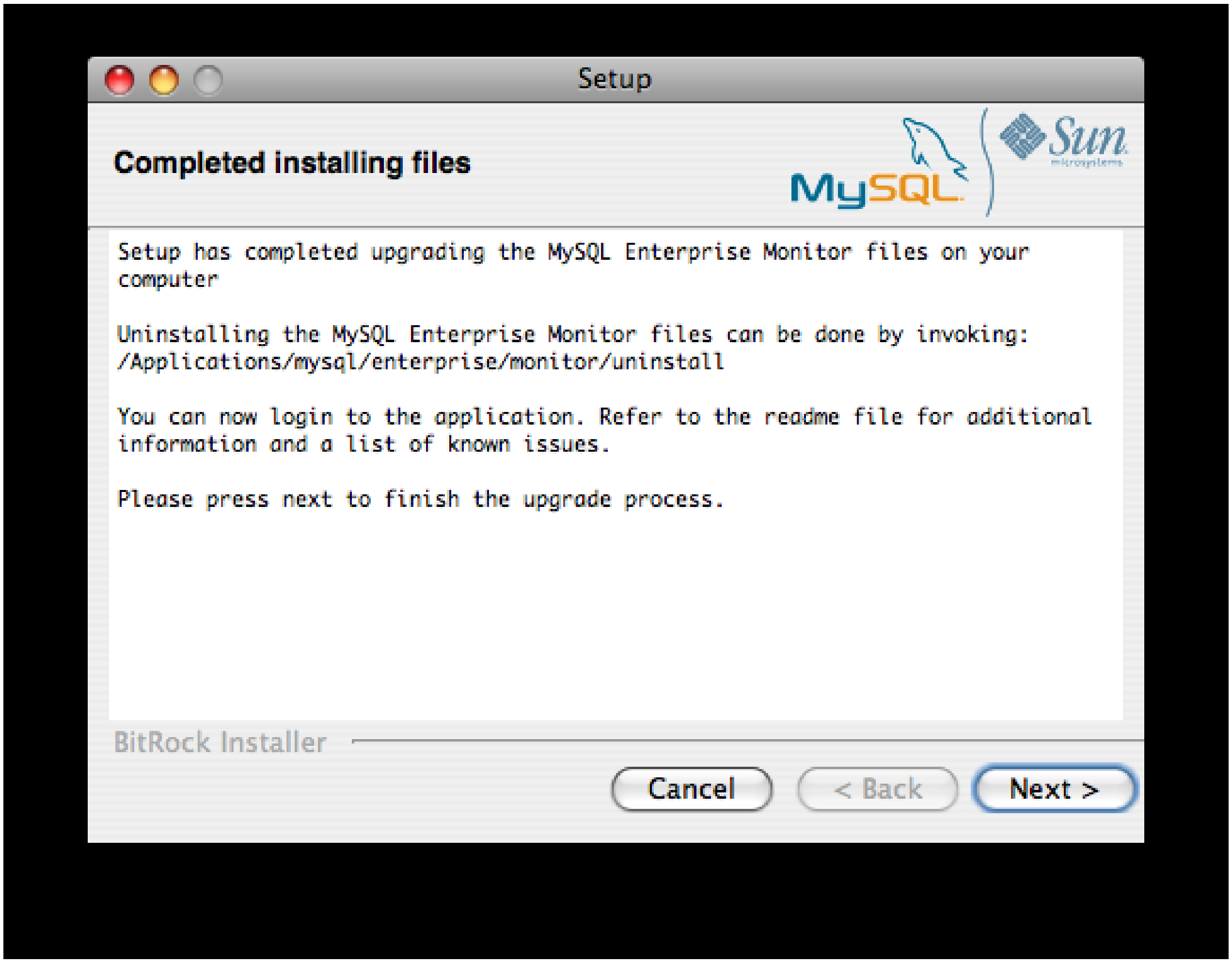
Specify the location of the backup (default is to use the `backup` directory within your installation directory). Note that backing up the database in addition to the main application will increase the installation time as the files have to be copied. The larger the size of your repository data, the longer the installation process will take.

6. Specify the Tomcat Server options. The Tomcat Server Port is the default port you will use to access the MySQL Enterprise Dashboard. If you want to support agents using SSL to communicate to MySQL Enterprise Service Manager, you must check the **Is SSL support required?**

7. Confirm that you want to continue the installation. Once installation has started, the backup of you existing application (and database) will start, although the process may take some time. Wait until the process completes.

8. Once the process has completed you will be provided with a notification of the installation process, including how to uninstall the application if you want to do so in the future. If any errors occurred, they will be reported here.

Figure 2.26. MySQL Enterprise Monitor: Server Update: Completed installing files



Completed installing files

Setup has completed upgrading the MySQL Enterprise Monitor files on your computer

Uninstalling the MySQL Enterprise Monitor files can be done by invoking:
/Applications/mysql/enterprise/monitor/uninstall

You can now login to the application. Refer to the readme file for additional information and a list of known issues.

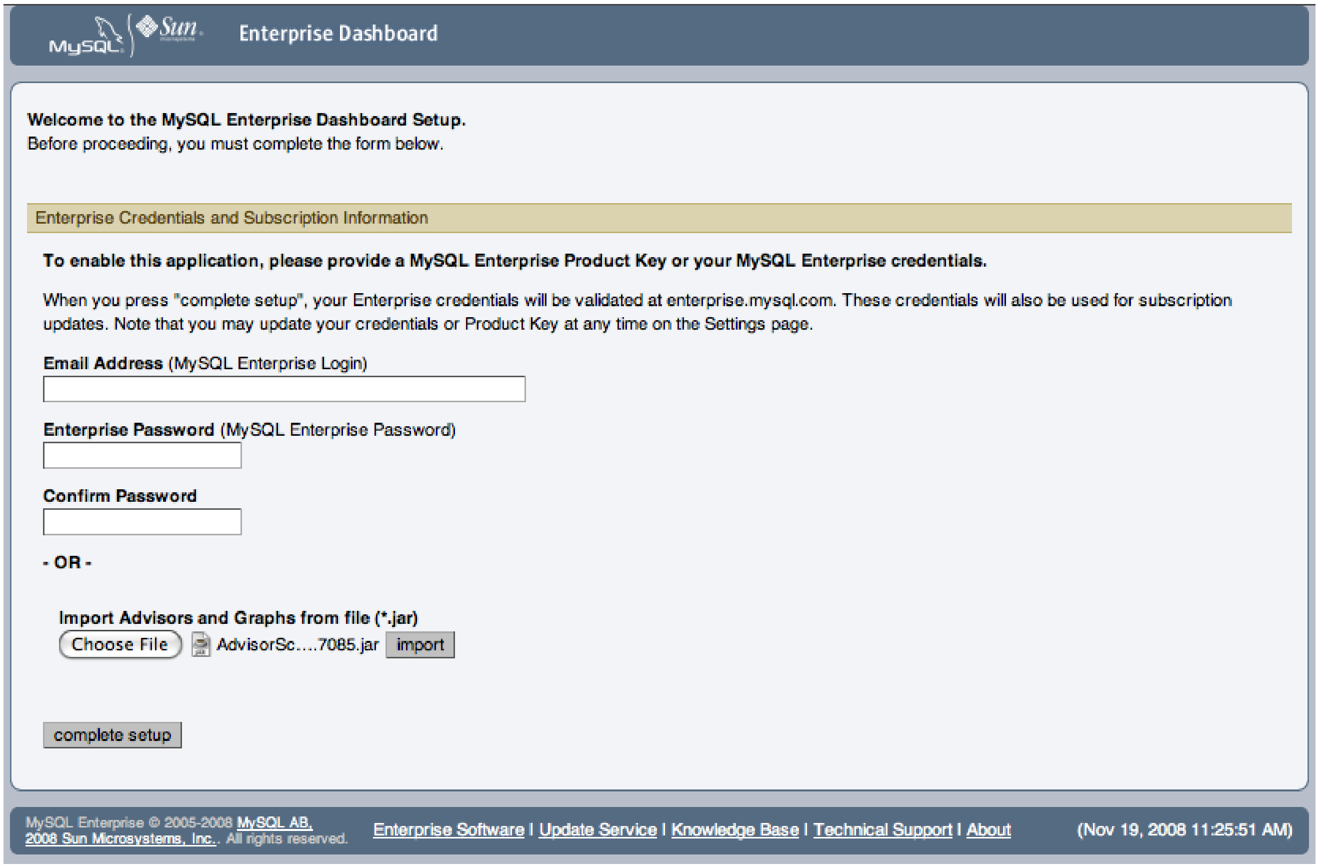
Please press next to finish the upgrade process.

9. The installation has now completed. You can automatically start the MySQL Enterprise Service Manager and view the attached Readme file by ensuring the checkboxes on this page are selected.

10. You can now quit the installer.

Once the installation has completed, the first time you login to MySQL Enterprise Dashboard you will be asked to provide your login credentials, if they do not already exist in the server configuration, or to provide a copy of the Advisor jar suitable for your MySQL Enterprise Service Manager version.

Figure 2.27. MySQL Enterprise Monitor: Server Update: Final Setup


MySQL Sun
Enterprise Dashboard
Choose File
AdvisorSc….7085.jar
import
complete setup
Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Welcome to the MySQL Enterprise Dashboard Setup.
Before proceeding, you must complete the form below.

Enterprise Credentials and Subscription Information

To enable this application, please provide a MySQL Enterprise Product Key or your MySQL Enterprise credentials.

When you press "complete setup", your Enterprise credentials will be validated at enterprise.mysql.com. These credentials will also be used for subscription updates. Note that you may update your credentials or Product Key at any time on the Settings page.

Email Address (MySQL Enterprise Login)

Enterprise Password (MySQL Enterprise Password)

Confirm Password

- OR -

Import Advisors and Graphs from file (*.jar)

MySQL Enterprise © 2005-2008 MySQL AB, 2008 Sun Microsystems, Inc.. All rights reserved.

(Nov 19, 2008 11:25:51 AM)

MySQL Enterprise Monitor has now been updated. You must update each of your agents to MySQL Enterprise Monitor Agent 2.0 to ensure that they are providing the correct information to MySQL Enterprise Service Manager

2.6.1.1.2. Upgrading to MySQL Enterprise Monitor Agent 2.0

To upgrade an agent you should use a `update` installer. This will migrate your configuration information, simplifying the upgrade process significantly.

> **Note**
>
> The agent log file, `mysql-monitor-agent.log`, if it exists, will be retained during the upgrade. A new log file, `mysql-monitor-agent.log` is used by MySQL Enterprise Monitor Agent 2.0.

The core sequence is the same on all platforms, the update process on Linux is shown below:

1. Start the update installer.

   ```
   shell> ./mysqlmonitoragent-2.0.0.7101-linux-glibc2.3-x86-32bit-update-installer.bin
   ```

2. Set the language for the installation process.

   ```
   Language Selection

   Please select the installation language
   [1] English
   [2] Japanese
   Please choose an option [1] :
   ```

3. Confirm or update the location of the installation directory of the previous version.

   ```
   ----------------------------------------------------------------------------
   ```

```
Welcome to the setup wizard for the MySQL Enterprise Monitor Agent Update

-----------------------------------------------------------------------------
Please specify the directory that contains the previous installation of
the MySQL Enterprise Monitor Agent

Installation directory [/opt/mysql/enterprise/agent]:
```

4. Specify whether you want to create a backup of the current application and configuration information, and if so, where the backup directory should be created.

```
-----------------------------------------------------------------------------
Current installation backup

Do you want to create a backup during the update process?

Backup the current installation [Y/n]: Y


Backup directory [/opt/mysql/enterprise/agent/patchbackup]:
```

5. You will be asked whether you want to enable the Query Analyzer. The Query Analyzer enables you to monitor the execution stateistics for individual queries executed through your MySQL servers. To enable, you must specify the proxy port, MySQL server and MySQL server port that you want to use. If you do not enable Query Analyzer now, you can enable it later. See Chapter 8, *The Query Analyzer Page*.

```
-----------------------------------------------------------------------------
Query Analyzer Configuration

MySQL Proxy enables query monitoring and analysis by listening on the port
specified below for client connections that are then passed through to a
backend MySQL database server. It is not needed for basic monitoring
functionality, but is required for query monitoring and analysis.

Visit the following URL for more information:
https://enterprise.mysql.com/docs/monitor/2.0/en/mem-query-analyzer.html



Enable Proxy (recommended) [Y/n]:


Proxy Port [4040]:

Backend Host: 127.0.0.1   (cannot be changed)

Backend Port: 3306   (cannot be changed)
```

6. You are now ready to complete the installation. Confirm that you want to continue.

```
-----------------------------------------------------------------------------
Setup is now ready to begin installing MySQL Enterprise Monitor Agent Update on your computer.

Do you want to continue? [Y/n]:

-----------------------------------------------------------------------------
Please wait while Setup installs MySQL Enterprise Monitor Agent Update on your computer.

 Installing
 0% ______________ 50% ______________ 100%
 #########################################

-----------------------------------------------------------------------------
Setup has finished installing MySQL Enterprise Monitor Agent Update on your computer.

Restart MySQL Enterprise Monitor Agent now [Y/n]:

View Readme File [Y/n]: n
```

Before connecting your MySQL Enterprise Monitor Agent to your MySQL server you must update the grants for the MySQL Enterprise Monitor Agent. Connect to the MySQL server and run this statement to update the required grants:

```
GRANT CREATE, INSERT
  ON mysql.*
  TO  'mysqluser'@'localhost'
  IDENTIFIED BY 'agent_password';
```

Replacing the `mysqluser` and `agent_password` parameters with the values used for connecting your agent to your MySQL server.

Once the update agent has communicated with the MySQL Enterprise Service Manager the core information about the agent and the MySQL server it is monitoring will be migrated to the new data format required by MySQL Enterprise Service Manager 2.0. To migrate the existing stored data, see Section F.9, “Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0”.

2.6.1.2. Unattended MySQL Enterprise Monitor Update

The options available when performing an unattended MySQL Enterprise Service Manager update are as follows:

```
--help                         Display the list of valid options

--version                      Display product information

--optionfile <optionfile>      Installation option file
                               Default:

--mode <mode>                  Installation mode
                               (Windows)Default: win32
                               (Unix)Default: gtk
                               (Mac OS X)Default: osx
                               (Windows)Allowed: win32 unattended
                               (Unix)Allowed: gtk text xwindow unattended
                               (Mac OS X)Allowed: osx text unattended

--debugtrace <debugtrace>      Debug filename
                               Default:

--installer-language <installer-language> Language selection
                                Default:
                                Allowed: en  jp

--installdir <installdir>      Previous Installation
                               Default:

--createDataBackup <createDataBackup>

                               Default: 1

--backupDir <backupDir>        Backup directory
                               Default:
```

The options for an unattended update of the agent differ only in that the `createDataBackup` option is replaced by `createBackup`.

If you did not install the MySQL Enterprise Service Manager to the default directory the `installdir` option must be specified. `mode` must also be specified when performing an unattended update. Otherwise, performing an unattended update is identical to the process described in Section 2.4, “Unattended Installation”.

2.6.2. Reinstalling MySQL Enterprise Monitor

In some cases you may want to reinstall MySQL Enterprise Monitor rather than updating your current installation. To reinstall rather than update MySQL Enterprise Monitor follow these steps:

1. Stop all the Monitor Agents
2. Run the `uninstall` programs for both the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent
3. Begin the new installation

To stop the Monitor Agents see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”

- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

Instructions for removing the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent are given in Section 2.7, “Uninstalling the MySQL Enterprise Monitor”.

2.6.3. Changing Your MySQL Enterprise Monitor Installation

This section describes the best practices to employ when changing your MySQL Enterprise Monitor installation.

2.6.3.1. Upgrading the Monitored MySQL Server

When upgrading a monitored MySQL server first stop the agent. To stop the agent see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

Stop the MySQL server and perform the upgrade. For instructions on stopping and restarting the MySQL service under Windows see Section 2.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows”.

To stop and restart the MySQL daemon under Unix and Mac OS X, see, Section 2.2.6, “Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X”.

Once the service/daemon is stopped you may upgrade your server. For instructions on upgrading your MySQL server see the reference manual pertaining to your server version. When the upgrade is complete restart the MySQL server.

> **Note**
>
> The agent's log file will show that the server was down.

2.6.3.2. Changing the Server That an Agent Monitors

You need not reinstall the MySQL Enterprise Monitor Agent to change the MySQL server that it monitors. It is possible to adapt an existing agent so that it monitors a different server.

To do this you must stop the monitor agent and then remove the server that it is monitoring. To stop the agent see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

For instructions on removing a server see, Section 4.3.3, “Removing a Server From the Dashboard”.

Once the agent is stopped and the server is removed from the Dashboard, changes may be made to the `mysql-monitor-agent.ini`, or the `agent-instance.ini` file within the agent `instances` instances directory. You can find the location of the directory by examining the content of the `mysql-monitor-agent.ini` and checking the value of the `mysqld-instance-dir` parameter.

If you want to make changes to the monitored MySQL server, edit the `agent-instance.ini` file. Change the `user`, `password`, `hostname`, and `port` values if required. For more information, see Section 2.3.6.2, “MySQL Server (`agent-instance.ini`) Configuration”.

To change other settings, such as enabling proxy support (required for Query Analyzer), the management host, or the port number used by the agent, modify the `mysql-monitor-agent.ini` file. For more information, see Section 2.3.6.1, “MySQL Enterprise Monitor Agent (`mysql-monitor-agent.ini`) Configuration”.

To restart the agent see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

> **Note**
>
> If you are adapting an existing agent to monitor a remote server make sure that the agent has the credentials for remote access and that the port on the remote MySQL server instance is open. For more information, see Section 2.3.6.4, “Configuring an Agent to Monitor a Remote MySQL Server”.
>
> If you experience difficulties starting the agent, check Section 2.3.7, “Troubleshooting the Agent”.

Log in to the Dashboard and you should find your new server in the `All Servers` group.

2.6.3.3. Temporarily Suspending the Agent

In some situations you may need to bring down a monitored server. When this is necessary, it is good practice to stop the agent first—doing so will avoid generating a “Server is unreachable” event.

For instance, suppose you need to stop the server to do a backup. The steps to follow are:

1. Stop the agent
2. Stop the service/daemon
3. Perform the backup
4. Restart the service/daemon
5. Restart the agent

To stop or start the agent see:

- Windows: see Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: see Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: see Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

To stop the MySQL service/daemon see the MySQL reference manual for your server version. You can find the manual online at http://dev.mysql.com/doc.

Follow these steps and there will be no “noise” associated with backing up your server. In contrast, if you leave the agent running while bringing down the server, you will generate a “Server is unreachable” event.

As an alternative to stopping the agent, you can change the logic associated with a rule. For instance, you could alter the threshold of the rule “Server is unreachable”:

```
%server.reachable% == THRESHOLD
```

to:

```
%server.reachable% == THRESHOLD  && CURTIME() NOT BETWEEN '22:00:00' AND '23:00:00'
```

This would effectively blackout the rule between 10 and 11 pm, during which time you could perform a backup.

For more information about editing rules see Section 5.3, “Editing Built-in Rules”. To blackout all events associated with a specific server or group of servers see Section 5.6, “Advisor Blackout Periods”.

2.7. Uninstalling the MySQL Enterprise Monitor

Removal of the MySQL Enterprise Monitor requires removal of the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent Service. In some circumstances, when running multiple agents on one machine for instance, you may not want to remove the entire MySQL Enterprise Monitor Agent Service but only a single monitored server.

2.7.1. Removing the MySQL Enterprise Monitor: Windows

Removing the MySQL Enterprise Service Manager

Remove the MySQL Enterprise Service Manager by going to the `Control Panel` and choosing `Add or Remove Programs`. Find the entry for `MySQL Enterprise Monitoring and Advisory Service` and remove it. During the uninstall process you will be given the option of saving existing data and log files. Choose this option if you plan to reinstall the MySQL Enterprise Monitor.

If you are not saving existing data, after MySQL Enterprise Service Manager has been removed you may delete the `C:\Program Files\MySQL\Enterprise\Monitor` directory.

> **Warning**
>
> If you chose not to remove existing data and log files when uninstalling MySQL Enterprise Service Manager do **not** remove the `C:\Program Files\MySQL\Enterprise\Monitor` directory. Doing so will delete these files.

If you added the Tomcat/Apache web server to the list of Windows firewall exceptions, remove this service by opening the `Windows Firewall` from the `Control Panel`. Choose the `Exceptions` tab and delete the `Tomcat/Apache` entry.

Removing MySQL Enterprise Monitor Services Only

When the MySQL Enterprise Service Manager is installed, the Tomcat/Apache and MySQL server services are started. It is possible to remove these services without also removing your MySQL Enterprise Service Manager installation. For more information about these services see, Section 2.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows”.

Do this by finding the `MySQL Enterprise Monitor` menu option and choosing `Services` and then `Uninstall MySQL Enterprise Monitor Services`. This will remove all the services associated with MySQL Enterprise Service Manager.

You can confirm that these services have been removed by checking services in the Microsoft Management Console Services window.

If you wish to reinstall these services you can do this by using the `Install MySQL Enterprise Monitor Services` menu option.

It is also possible to remove services using the `mysqlmonitorctl.bat` file found in the `C:\Program Files\MySQL\Enterprise\Monitor` directory. To see the available options, go to the command line and type: **`myqlnetworkctrl help`**. This batch file is discussed in more detail in Section 2.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows”.

Removing the Monitor Agent

To remove the Monitor Agent itself, open the `Control Panel` and choose `Add or Remove Programs`. Find the entry for `MySQL Enterprise Monitor Agent` and remove it. This will execute the uninstall program located in the `C:\Program Files\MySQL\MySQL\Enterprise\Agent` directory.

> **Warning**
>
> If you are running more than one agent on the same machine and wish to remove only one of the agents, do **not** remove the `MySQL Enterprise Monitor Agent` entry from the `Add or Remove Programs` menu. To remove a single agent see Removing a Single Agent [72]

After removing the Monitor Agent you may also need to remove the directories, `C:\Program Files\MySQL\Enterprise` and `C:\Program Files\MySQL\Enterprise\Agent`.

Removing the Monitor Agent in this fashion will remove the default service. However, if you are running additional Monitor Agents as described in Section 2.3.6.2, “MySQL Server (`agent-instance.ini`) Configuration”, you will have to remove those agents manually. See the next section for instructions on doing this.

Removing a Single Agent

If you are running more than one agent on the same machine and wish to remove only one of the agents, do **not** remove the `MySQL Enterprise Monitor Agent` entry from the `Add or Remove Programs` menu. To remove a single agent and leave other agents intact follow these steps:

1. Stop the agent
2. Confirm the location of the log files
3. Remove the agent as a service
4. Remove/Archive the associated files

It is best to stop the agent before removing it; for instructions on stopping an agent see, Section 2.3.5.1, “Starting/Stopping the Agent on Windows”.

You can confirm the location of the agent log files by checking the `ini` file. For more information on this topic see Section 2.3.6.1, “MySQL Enterprise Monitor Agent (`mysql-monitor-agent.ini`) Configuration”.

Go to the command line and remove the MySQL Enterprise Monitor Agent as a Windows service by typing:

```
shell> sc delete AgentName
```

You can confirm that the agent has been removed by checking the Microsoft Management Console Services window. There should no longer be an entry for the removed agent.

You should also remove or archive any log or configuration files associated with this agent. If you have installed any additional agents, remove them in the same fashion.

2.7.2. Removing the MySQL Enterprise Monitor: Unix

Removing the MySQL Enterprise Service Manager

To remove the MySQL Enterprise Service Manager, find the `uninstall` file in the `/opt/mysql/enterprise/monitor` directory.

Execute this file by typing:

```
shell> ./uninstall
```

During the uninstall process you will be given the option of saving existing data and log files. Choose this option if you plan to reinstall the MySQL Enterprise Monitor.

If you are not saving existing data, after uninstalling the MySQL Enterprise Service Manager you may remove the `/opt/mysql/enterprise/monitor` directory.

> **Warning**
>
> If you chose not to remove existing data and log files when uninstalling the MySQL Enterprise Monitor do **not** remove the `/opt/mysql/enterprise/monitor` directory; doing so will delete these files.

On Red Hat Enterprise Linux 4 and Fedora Core 4, the uninstall script may not stop the Tomcat server. Do this manually if necessary. To do this see, Section 2.2.6, “Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X”.

There may be other Java processes running on your system. Be careful not to accidentally stop them.

On some Unix platforms, inluding HP-UX, you may have to manually delete the `uninstall` application and the installation directory after you have execute the uninstall process.

Removing the Monitor Agent

Prior to removal of the Monitor Agent Service you should stop any agents. Do this by changing to the `init.d` directory and issuing the command, `./mysql-monitor-agent stop`.

You will find the `uninstall` file in the `/opt/mysql/enterprise/agent` directory. Execute this file by navigating to this directory and typing:

```
shell> ./uninstall
```

After uninstalling the Monitor Agent you may remove the `/opt/mysql/enterprise/agent` directory.

Removing the Monitor Agent in this fashion will remove the default service, and all the configuration files for different instances.

Removing a Single Agent

If you are running more than one agent on the same machine and wish to remove only one of the agents, do **not** run the uninstall program. To remove a single agent and leave other agents intact follow these steps:

1. Stop the agent
2. Confirm the location of the log files
3. Remove the agent as a service
4. Remove/Archive associated files

It is best to stop the agent before removing it; for instructions on stopping an agent see Section 2.3.5.3, “Starting/Stopping the Agent on Unix”.

You can confirm the location of the agent log files by checking the `ini` file. For more information on this topic see Section 2.3.6.1, “MySQL Enterprise Monitor Agent (`mysql-monitor-agent.ini`) Configuration”.

You may then remove the agent as a daemon by removing its entry in the `init.d` directory. You should also remove or archive any log or configuration files associated with this agent.

If you have installed any additional agents, remove them in the same fashion.

2.7.3. Removing the MySQL Enterprise Monitor Mac OS X

Removing the MySQL Enterprise Service Manager

To remove the MySQL Enterprise Service Manager, run the `uninstall.app` located in the `/Applications/mysql/enterprise/monitor/` directory, or the root directory of your MySQL Enterprise Service Manager installation.

During the uninstall process you will be given the option of saving existing data and log files. Choose this option if you plan to reinstall the MySQL Enterprise Monitor.

If you are not saving existing data, after uninstalling the MySQL Enterprise Service Manager you may remove the `/Applications/mysql/enterprise/monitor` directory.

> **Warning**
>
> If you chose not to remove existing data and log files when uninstalling the MySQL Enterprise Monitor do not remove the /Applications/mysql/enterprise/monitor directory; doing so will delete these files.

Removing the Monitor Agent

Prior to removal of the MySQL Enterprise Monitor Agent you should stop any agents. Do this by changing to the `init.d` directory and issuing the command:

```
shell> ./mysql-monitor-agent stop
```

Run the `uninstall.app` file located in the `/Applications/mysql/enterprise/agent` directory.

After uninstalling the MySQL Enterprise Monitor Agent you may remove the `/Applications/mysql/enterprise/agent` directory.

Removing the MySQL Enterprise Monitor Agent in this fashion will remove the default service, and all the configuration files for different instances.

Removing a Single Agent

If you are running more than one agent on the same machine and wish to remove only one of the agents, do not run the uninstall program.

To remove a single agent and leave other agents intact follow these steps:

1. Stop the agent
2. Confirm the location of the log files
3. Remove the agent as a daemon
4. Remove/Archive associated files

It is best to stop the agent before removing it; for instructions on stopping an agent see Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”.

You can confirm the location of the agent log files by checking the ini file. For more information on this topic see Section 2.3.6.1, “MySQL Enterprise Monitor Agent (`mysql-monitor-agent.ini`) Configuration”.

You may then remove the agent as a daemon by removing its entry in the `init.d` directory.

You should also remove or archive any log or configuration files associated with this agent.

If you have installed any additional agents, remove them in the same fashion.

Chapter 3. MySQL Enterprise Dashboard

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

The purpose of the MySQL Enterprise Dashboard is to provide you with information about your MySQL servers. It provides a list of the latest MySQL Enterprise Advisor reports, server status information, MySQL Enterprise alerts, and updated views of monitored MySQL servers. The `Monitor` screen gives a quick overview of the status of your MySQL servers.

The MySQL Enterprise Dashboard interface is provided through a standard web browser. For more details on the supported browsers, see Section F.2, “Supported Browsers”.

You can open the Dashboard and connect to the MySQL Enterprise Dashboard either using the standard HTTP protocol, or using an SSL connection (HTTPS). By default, you can connect to MySQL Enterprise Dashboard using `http://host:18080`, where `host` is the hostname or IP address of the MySQL Enterprise Service Manager. The default SSL URL is `http://host:18443`. These values can be overridden during installation. You can check the configured values by looking at the `configuration_report.txt` file within your installation directory.

> **Note**
>
> If you are connecting using SSL, the built-in certificate has been self-signed and may be highlighted as 'unsafe' within the browser on initial connection. You should add an exception for the certificate on this server to prevent problems accessing the site.

The Monitoring page provides an instant health check for all of the MySQL servers across the enterprise.

From this page users can:

- View monitoring data and all critical MySQL Advisor Rule violations for all or selected servers.
- Close and annotate MySQL Advisor Rule violations.
- Quickly determine if there is a Monitor Agent that is not communicating with the Service Manager.
- Quickly determine if there is a server that is in trouble or completely down.
- View indicator value graphs for key MySQL and operating system (OS) level metrics. Graph presentation will default to a thumbnail view but will open into a larger image upon being clicked.

The monitored server or servers are displayed in a tab on the left known as the `Server Tree`. You can navigate to a number pages that provide more detailed information. These pages include:

- `Monitor`: The overview page providing you with a quick summary of the servers, their status, events, availability and load. The remainder of this chapter details the contents of this page.
- `Advisors`: Shows the various advisors configured in your installation and allows you to schedule their execution on different servers, apply and manage rules and manage the advisor installation itself. For more information, see Chapter 5, *The Advisors Page*.
- **EVENTS**: Provides an interface into the event system that highlights specific issues and problems on your monitored servers. For more information on using Events, see Chapter 6, *The Events Page*.
- `Query Analyzer`: Interfaces to the query monitoring system that can be used to monitor and track the individual queries that are being executed on a system and help to highlight problem queries that may need optimization or that may be affecting server load. For more information, see Chapter 8, *The Query Analyzer Page*.
- `Graphs`: Enables you to view and configure a number of individual graphcs covering a range of different statistics. For more details on how to view and use these graphs, see Chapter 7, *The Graphs Page*.
- `Replication`: Provides information on the status and structure of your servers that are using replication. This page is only available if you have a suitable subscription level. For more information, see Chapter 9, *The Replication Page*.

- `Settings`: Controls the settings for the server, including email configuration, passwords, and server and user management. For more information, see Chapter 4, *The Settings Page*.

Graphs are shown in the center of the page beneath the tabs. If applicable, you'll also find a list of critical events.

On the right is the color-coded `Heat Chart`, showing the advisors that are installed by default. The `Heat Chart` shows the most important advisors, allowing a quick overview of the state of your servers. You may open the `Heat Chart` in its own window by clicking the `Standalone Heat Chart` link. If applicable, you'll also find a list of critical events.

The `Show/Hide Legend` link toggles display of the key to the icons used in the `Heat Chart`.

> **Note**
>
> Find colorblind-accessible icons in the `alternate` directory. On Linux this directory is immediately below the `/monitor/apache-tomcat/webapps/ROOT/web/resources/images/` directory. These images are stored in the same directory on Windows. To use them, backup the originals and then copy and paste the alternate set into the `images` directory.

If a specific server is selected in the `Server Tree` details about this server are shown beneath the legend in the `Meta Info` area. The information shown in this area is the host name, the MySQL version number, the number of scheduled rules, the operating system, and the CPU.

The `Meta Info` section also shows how long the agent has been running, when it last contacted the MySQL server it is monitoring, and the last time the agent contacted the dashboard. Mouse over the date shown beside **Up Since** and a pop-up box displays the time that has elapsed since the server instance was last started. You can also mouse over the **Last MySQL Contact** and the **Last Agent Contact** dates.

> **Note**
>
> In the case of remote monitoring, the agent runs on a different machine than the MySQL server that it is monitoring. The `Hostname`, `MySQL`, and `Rules` information applies to the system being monitored. The `OS` and `CPU` information applies to the machine on which the agent is running. For more information about remote monitoring see, Section 2.3.6.4, “Configuring an Agent to Monitor a Remote MySQL Server”.

The top of the screen shows the refresh cycle and `Help` and `Log Out` links. Click the `Help` link to open the documentation in a separate browser window. Choose `Log Out` if you wish to leave the Dashboard or to log in as a different user. Different refresh rates are available from the drop-down listbox.

In the footer are external links to MySQL Enterprise and information about the current user. Users can remain connected to the Dashboard and update their subscription, use the Enterprise Knowledge Base, and contact technical support. Your subscription information is also displayed here, showing the number of days remaining and the number of licenses. The number of licenses indicates to the number of machines that may be monitored; any number of MySQL servers may be running on a specific machine.

The footer also contains a link to the `Settings` page. If your subscription is current it reads **Subscription is up-to-date. More info....** For more information about the `Settings` page see Section 4.7, “The Product Information Screen”.

3.1. The Server Tree

The tab on the left displays the Server tree. By default the first group of servers is selected. This selection determines the information shown on the `Monitor` page.

If a server group is selected, the information presented on the `Monitor` page is aggregate information for this group; if only one server is selected the information applies to that server only.

Change your server selection and the information shown in the graphs and in the `Heat Chart` changes.

> **Note**
>
> For more information about server groups see, Section 4.3.2, “Grouping Servers”.

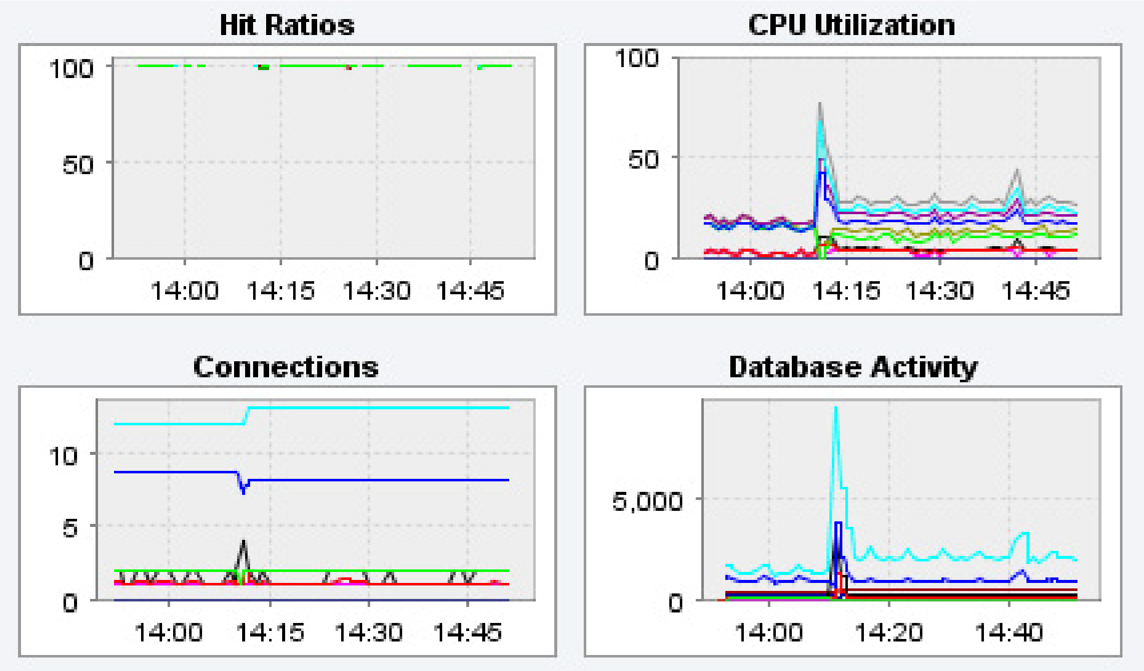
The individual server, or server group, selected in the Server Tree also determines what information appears when the `Advisors` tab or the **Events** tab is selected.

The Server Tree presents an easy way to navigate to different groups or to specific servers.

3.2. The Server Graphs and Critical Events

The center of the `Monitor` page gives a visual representation of the state of your servers.

Figure 3.1. MySQL Enterprise Dashboard: The Graphs Screen



The graphs present information about the currently selected server or server group. The default graphs show the hit ratios, CPU utilization, connections, and database activity for a specific interval.

To set the interval click the `configure graphs` link immediately below the graphs. This opens a dialog box where you can choose the default interval for the x-axis of the graphs. Defining a shorter or longer interval gives you a shorter or longer term view of server activity. The thumbnail and full-size graph dimensions can also be adjusted from this dialog box. Save any changes that you have made and the values chosen will be the defaults whenever you log in.

You can also choose the default graphs shown on the `Monitor` page. To do this click the `edit favorites` link and choose the graphs you want from the drop-down list box. To choose contiguous graphs, hold down the **Shift** key and click the desired graphs. For a noncontiguous selection, click the desired graphs while holding down the **Ctrl** key. The maximum number of graphs that can be displayed on the `Monitor` page is six. Save your changes and these will be the default graphs whenever you log in.

Color coding helps distinguish different aspects of each graph. With `Database Activity` for example, you can readily distinguish `SELECT` statements from database insertions.

Clicking a graph opens a detailed view with **GRAPH DISPLAY** and **CONFIGURE** tabs. Choose the `Configure` tab to temporarily change the way that a graph displays. Changes made from this tab only apply to the standalone graph while it is open. Persistent changes are made as described above.

Dismiss the enlarged graph by clicking the HIDE button.

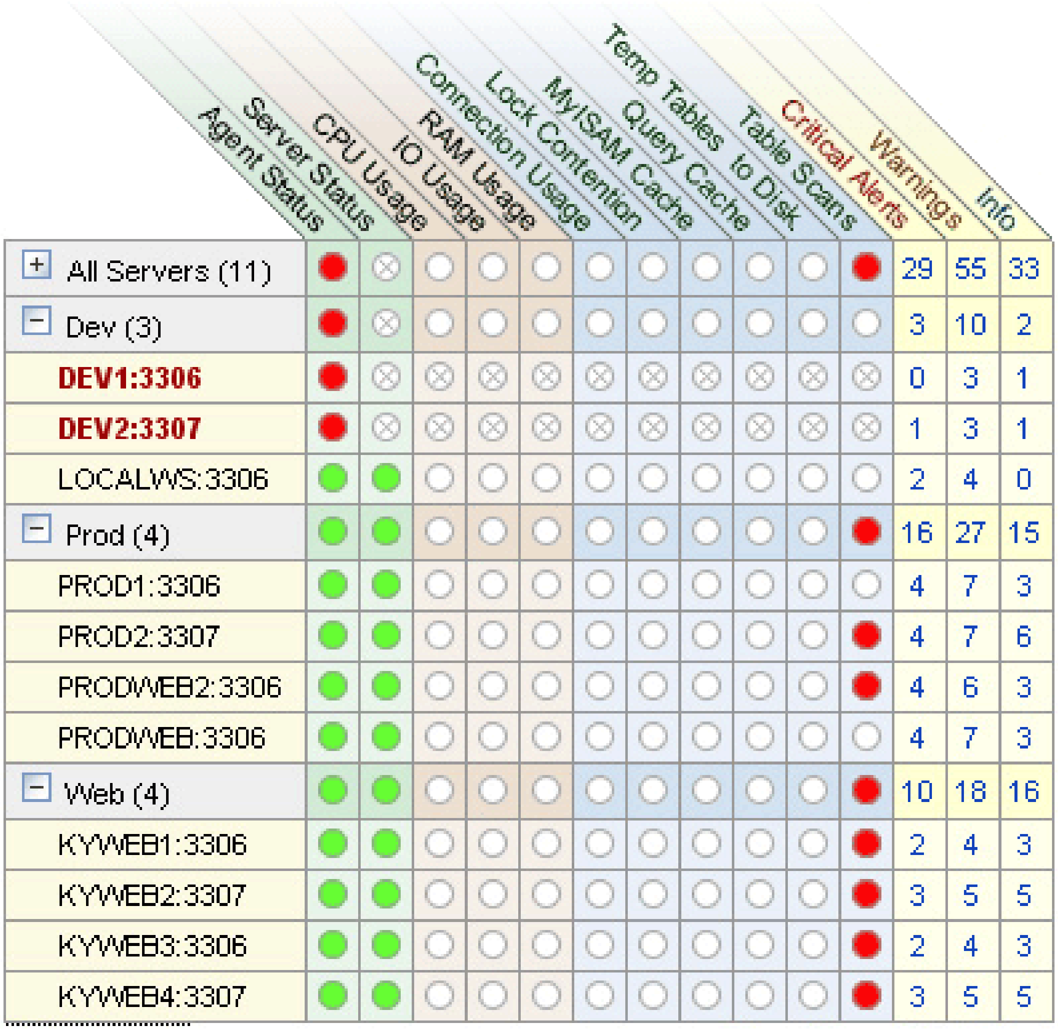
Alerts from the event system will be displayed immediately below the charts. For more information on the events display, see Chapter 6, *The Events Page*.

3.3. The Heat Chart

The `Heat Chart` is found on the right side of the `Monitor` page and shows the status of critical rules. Monitored servers are organized by groups. To view the status of a specific server, click the + button next to the appropriate server group.

Whenever a new agent contacts the Service Manager for the first time, all the rules in the Heat Chart Advisor are automatically activated. These Advisors monitor the status of the server and agent, critical operating system indicators, and important events related to your MySQL servers. An example follows.

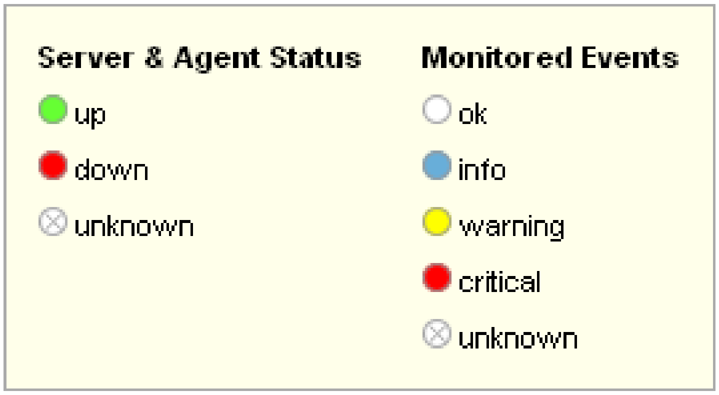
Figure 3.2. MySQL Enterprise Dashboard: The Heat Chart



	Agent Status	Server Status	CPU Usage	IO Usage	RAM Usage	Connection Usage	Lock Contention	MyISAM Cache	Query Cache	Temp Tables to Disk	Table Scans	Critical Alerts	Warnings	Info
+ All Servers (11)	🔴	⊗	○	○	○	○	○	○	○	○	🔴	29	55	33
− Dev (3)	🔴	⊗	○	○	○	○	○	○	○	○	○	3	10	2
DEV1:3306	🔴	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	0	3	1
DEV2:3307	🔴	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	1	3	1
LOCALWS:3306	🟢	🟢	○	○	○	○	○	○	○	○	○	2	4	0
− Prod (4)	🟢	🟢	○	○	○	○	○	○	○	○	🔴	16	27	15
PROD1:3306	🟢	🟢	○	○	○	○	○	○	○	○	○	4	7	3
PROD2:3307	🟢	🟢	○	○	○	○	○	○	○	○	🔴	4	7	6
PRODWEB2:3306	🟢	🟢	○	○	○	○	○	○	○	○	🔴	4	6	3
PRODWEB:3306	🟢	🟢	○	○	○	○	○	○	○	○	○	4	7	3
− Web (4)	🟢	🟢	○	○	○	○	○	○	○	○	🔴	10	18	16
KYWEB1:3306	🟢	🟢	○	○	○	○	○	○	○	○	🔴	2	4	3
KYWEB2:3307	🟢	🟢	○	○	○	○	○	○	○	○	🔴	3	5	5
KYWEB3:3306	🟢	🟢	○	○	○	○	○	○	○	○	🔴	2	4	3
KYWEB4:3307	🟢	🟢	○	○	○	○	○	○	○	○	🔴	3	5	5

To interpret the Heat Chart see the following legend.

Figure 3.3. MySQL Enterprise Dashboard: The Heat Chart Legend



Server & Agent Status

🟢 up

🔴 down

⊗ unknown

Monitored Events

○ ok

🔵 info

🟡 warning

🔴 critical

⊗ unknown

The status `unknown` will typically apply when an agent is down and can no longer report the status of the server that it is monitoring. The status `unknown` may also apply if the data collection that should be collected is not available on the server being monitored.

You may open the Heat Chart in its own browser window by clicking the `Standalone Heat Chart` link immediately below the `Heat Chart` on the left. If you like, the refresh rate can be set to a different rate than the setting on the `Monitor` page.

In addition to showing the most important advisors, the `Heat Chart` also has columns that display the number of critical, warning, and informational alarms. Clicking the hyperlink in any one of these columns takes you to the `Event` screen, which gives more detailed information. For more information about events see, Chapter 6, *The Events Page*.

When the Dashboard is first installed no notification groups are associated with the Advisors shown in the Heat Chart. For more information on this topic see, Section 2.2.7.3, “Installing Advisors After Initial Log-in” and, Section 4.5, “Manage Notification Groups”.

Chapter 4. The Settings Page

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

Upon initial installation you may have configured your MySQL Enterprise credentials and also outgoing email settings. This section explores the configuration settings in more detail, and also shows how to manage servers, users, notification groups, Simple Network Management Protocol (SNMP) traps, log files, and the product information screen.

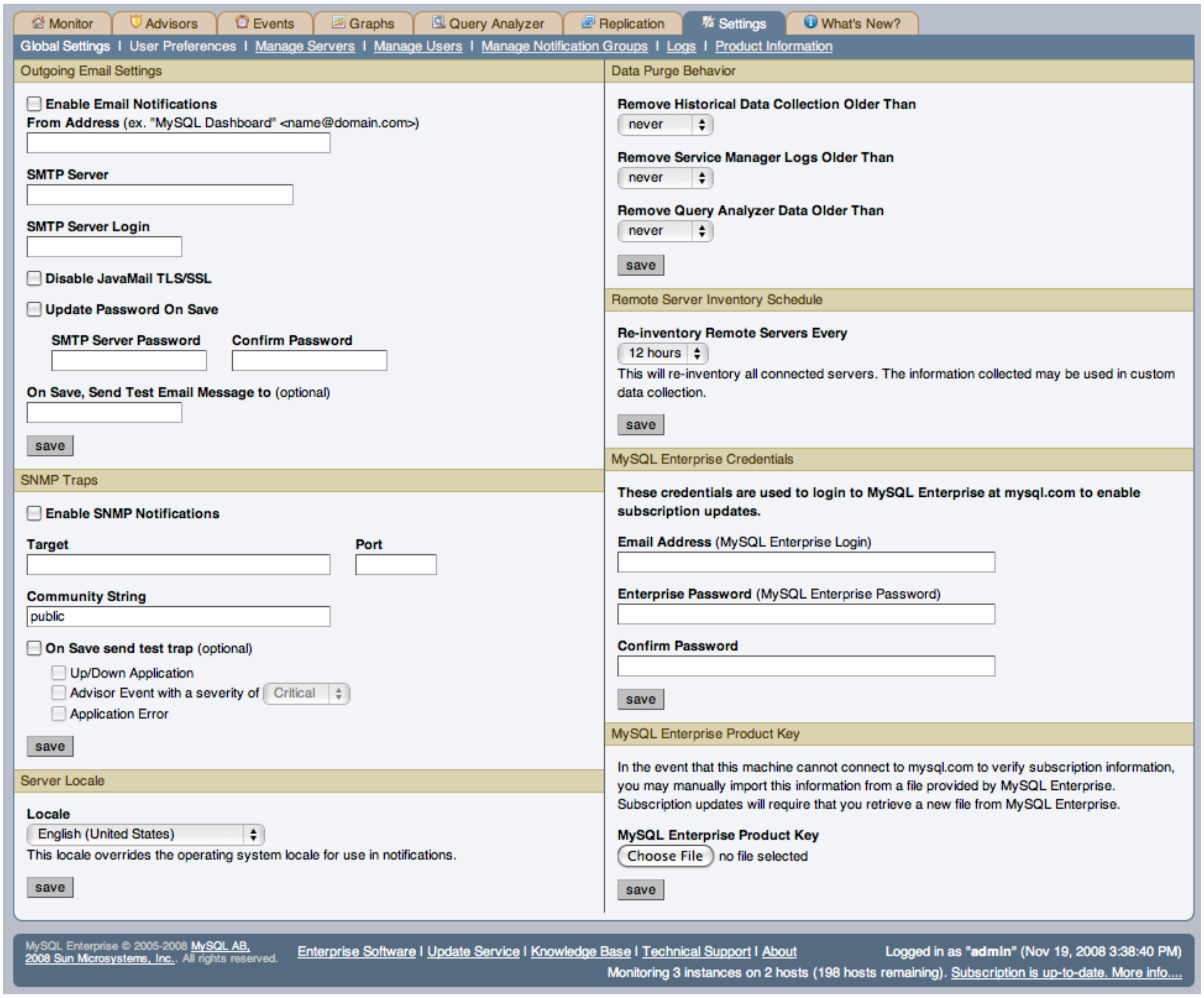
Knowledge of server management is a prerequisite for properly configuring advisors—the subject of Chapter 5, *The Advisors Page*.

To get to the `Settings` page open the Dashboard and choose the `Settings` tab.

4.1. Global Settings

The **Global Settings** control the main confirguration parameters for the entire MySQL Enterprise Monitor system, including your email notifications, data purge, and Enterprise website credentials.

Figure 4.1. MySQL Enterprise Dashboard: Settings


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New?
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information

Outgoing Email Settings

□ Enable Email Notifications

From Address (ex. "MySQL Dashboard" <name@domain.com>)

SMTP Server

SMTP Server Login

□ Disable JavaMail TLS/SSL

□ Update Password On Save

SMTP Server Password   Confirm Password

On Save, Send Test Email Message to (optional)

save

SNMP Traps

□ Enable SNMP Notifications

Target   Port

Community String: public

□ On Save send test trap (optional)

- □ Up/Down Application
- □ Advisor Event with a severity of Critical
- □ Application Error

save

Server Locale

Locale: English (United States)

This locale overrides the operating system locale for use in notifications.

save

Data Purge Behavior

Remove Historical Data Collection Older Than: never

Remove Service Manager Logs Older Than: never

Remove Query Analyzer Data Older Than: never

save

Remote Server Inventory Schedule

Re-inventory Remote Servers Every: 12 hours

This will re-inventory all connected servers. The information collected may be used in custom data collection.

save

MySQL Enterprise Credentials

These credentials are used to login to MySQL Enterprise at mysql.com to enable subscription updates.

Email Address (MySQL Enterprise Login)

Enterprise Password (MySQL Enterprise Password)

Confirm Password

save

MySQL Enterprise Product Key

In the event that this machine cannot connect to mysql.com to verify subscription information, you may manually import this information from a file provided by MySQL Enterprise. Subscription updates will require that you retrieve a new file from MySQL Enterprise.

MySQL Enterprise Product Key

Choose File no file selected

save

MySQL Enterprise © 2005-2008 MySQL AB, 2008 Sun Microsystems, Inc.. All rights reserved.

Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "admin" (Nov 19, 2008 3:38:40 PM)

Monitoring 3 instances on 2 hosts (198 hosts remaining). Subscription is up-to-date. More info....

The **Global Settings** page is divided into a number of different sections:

- **Outgoing Email Settings**

  Configures the settings for email notifications by MySQL Enterprise Service Manager. You must configure the **From Address SMTP Server** settings. If your server requires authorization, complete the necessary server login details, and whether SSL is required.

  You can test your configuration immediately by adding an email address to the **On Save, Send Test Email Message to** box.

  For more information about `Outgoing Email Settings` see, Section 2.2.7.5, "Outgoing Email Settings".

- **SNMP Traps**

  The **SNMP Traps** section of the **Global Settings** tab allows you to enable Simple Network Management Protocol so that your Network Management System (NMS) can handle events created by the MySQL Enterprise Monitor. SNMP notifications are off by default. You can enable them by clicking the **Enable SNMP Notifications** checkbox.

  MySQL Enterprise Service Manager includes support for both SNMPv1 and SNMPv2. You can select the SNMP version to use by clicking the corresponding **Use SNMPv1** or **Use SNMPv2** radio button.

  In the **Target 1** or **Target 2** text box enter the IP address or hostnames of your NMS listeners. The port number defaults to the well-known SNMP port, `162`. If you are not using this port, enter the port that your Network Management System is listening on.

Enter the appropriate community string in the `Community String` text box. The default value for this string is `public`.

For the trap address for Advisor traps, you can optionally elect to use the address of the agent that triggered the alert, rather than the address of the MySQL Enterprise Service Manager. To do this, click the **Use the remote MySQL agent host IP address as the SNMP trap agent address for Advisor traps**. For internally generated SNMP traps (i.e. with MySQL Enterprise Service Manager) you can also specify an alternate address by putting the IP address in the **SNMP trap agent address for internally generated traps** box.

To ensure that the target you have specified is valid, check the `On Save, Send Test Trap` check box. The remaining check boxes help you to configure how your NMS responds to MySQL Enterprise Monitor. Check the **Up/Down Application** check box to configure NMS for starting up or shutting down the MySQL Enterprise Monitor. For configuration of advisor events choose a level of severity and check the **Advisor event with the severity of *`Critical`*** check box. Finally, choose the **Application Error** check box to configure NMS to support application error traps. Be sure to save your settings before exiting.

If you wish to enable SNMP traps globally, check the `Enable SNMP Notifications` checkbox. To enable SNMP traps only for specific rules run against specific servers or server groups leave this checkbox unchecked—enabling specific SNMP traps is done as rules are scheduled. For instructions on doing this see Section 5.2, “Scheduling Rules”.

The Management Information Base (MIB) file associated with SNMP trapping is called `MONITOR.MIB`. For the location this file see Section C.4, “The Management Information Base (MIB) File”.

> **Note**
>
> The MIB file supplied with MySQL Enterprise Service Manager is not directly compatible with OpenNMS. For a sample of the modifications that need to be made, please see the comments for Bug#41714.

- **Server Locale**

  The `Server Locale` setting determines the language of notification for the following items:

  - Email notifications
  - SNMP traps
  - The naming conventions for shared resources such as a replication group name prefix

  The initial value in this drop down list box is the locale for the OS on which the Dashboard is running.

- **Data Purge Behavior**

  The `Data Purge Behavior` section of the `Global Preferences` page lets you remove old log files and also old data from the repository. The default purge interval is `never`. If you wish to purge data, change this setting by choosing from the drop-down list. Choosing `52 weeks`, for example, will remove all data that is older than a year.

  > **Warning**
  >
  > Purging data will permanently remove information from the repository. Since events are derived from data contained in the repository, they will be purged along with the data.
  >
  > Ensure that there is adequate disk space for the repository. If you are monitoring numerous servers and running many rules the size of the repository can increase rapidly. Choose purge behavior accordingly.
  >
  > The default value for purging, `never`, is the safest option. However, please choose a purge setting that makes sense for your environment.

  > **Note**
  >
  > The purge process is started approximately once every minute. If you change the purge duration from a larger timespan to a smaller one, the data may start to be purged immediately.

  You can configure the data purge behavior for a number of different systems individually:

  - **Remove Historical Data Collection Older Than** configures the duration that the main data about your servers is retained. This includes all data collections, including CPU, memory and connections and activity statistics.

  - **Remove Service Manager Logs Older Than** configures the duration that the main MySQL Enterprise Service Manager logs are retained.

  - **Remove Query Analyzer Data Older Than** configures the duration that the query analyzer statistics and information about individual queries is retained.

  Notes for setting purge behavior:

  - Purging can be carried out manually by enabling the `innodb_file_per_table` for the repository database and then using an `OPTIMIZE TABLE` operation to reclaim space from deleted rows in the table.

  - If you set the purge value to a very low timespan value when the previous setting was quite high then the space used for the data that was purged will not be reclaimed from the InnoDB tablespaces. You can do this by running `OPTIMIZE TABLE` on the MySQL tables for MySQL Enterprise Service Manager to reclaim the space from the purged rows.

- **Remote Server Inventory Schedule**

  MySQL Enterprise Monitor keeps track of all the databases and tables in a server, as well as the amount of RAM, disk space, and other items. A re-inventory updates this information in case you have added or dropped databases and tables. Depending upon the configuration of your system, this operation can tax resources. If you are monitoring many remote servers this is an operation you may want to perform in off-peak hours only.

- **MySQL Enterprise Credentials**

  You can specify the credentials for logging into the MySQL Enterprise Website. These should match the user name and password that you have registered with MySQL for your enterprise subscription.

  > **Note**
  >
  > Only administrators can change the `MySQL Enterprise Credentials` section or enter a product key; for other users, this section does not show up in the interface. For more information about different users and their rights see Section 4.4, “Managing Users”. Specifying incorrect credentials results in the error message, “Your credentials do not appear to be valid.”

- **MySQL Enterprise Product Key**

  You may update your `MySQL Enterprise Product Key`. If you do not have access to the Internet from the Dashboard, this provides an alternate way to update or activate the MySQL Enterprise Monitor.

  To enter your product key first download it from the MySQL Enterprise website. Copy the key to a location accessible from the Dashboard. Use the Browse button to locate the key and then press the Save button.

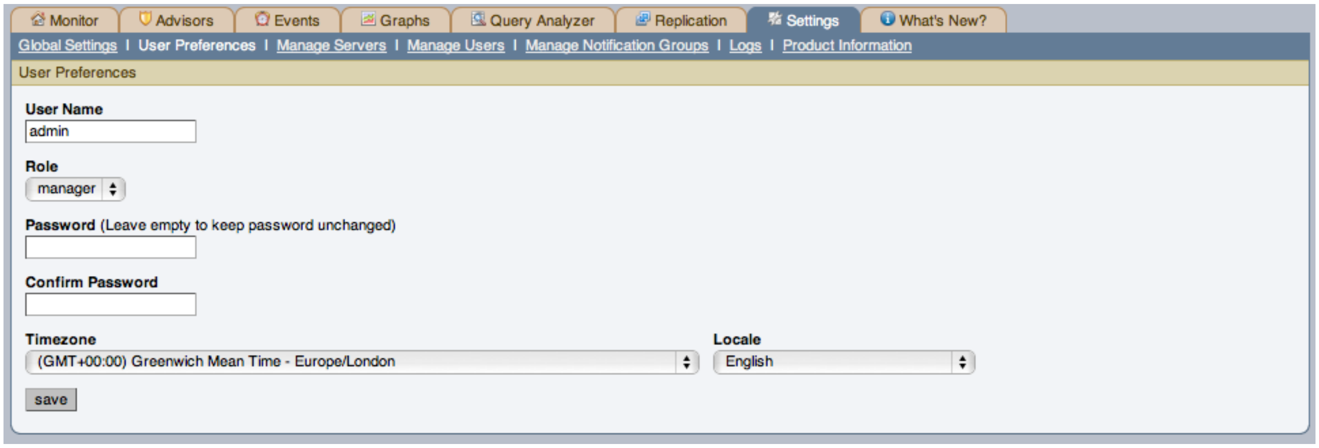
  If you wish to switch from using your MySQL Enterprise credentials to using a product key to update MySQL Enterprise Monitor, you must first clear your credentials. Do this by removing the email address from the **MySQL Enterprise Credentials** section and then clicking the Save button. You may then enter and save your MySQL Enterprise product key.

  > **Note**
  >
  > Only administrators can change the `MySQL Enterprise Credentials` section or enter a product key; for other users, this section does not show up in the interface. For more information about different users and their rights see Section 4.4, “Managing Users”. Specifying incorrect credentials results in the error message, “Your credentials do not appear to be valid.”

4.2. User Preferences

On this page users can change their passwords, user names, and locale information.

Figure 4.2. MySQL Enterprise Dashboard: User Preferences


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New?
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information
save

User Preferences

User Name: admin

Role: manager

Password (Leave empty to keep password unchanged)

Confirm Password

Timezone: (GMT+00:00) Greenwich Mean Time - Europe/London

Locale: English

Change your password by entering a new value into the **Password** text box. To change your user name enter a new value into the **Username** text box. Click the SAVE USER PROPERTIES button to commit this change.

You may also adjust your time zone and locale information from this page. The settings on this page apply only to the user who is currently logged in.

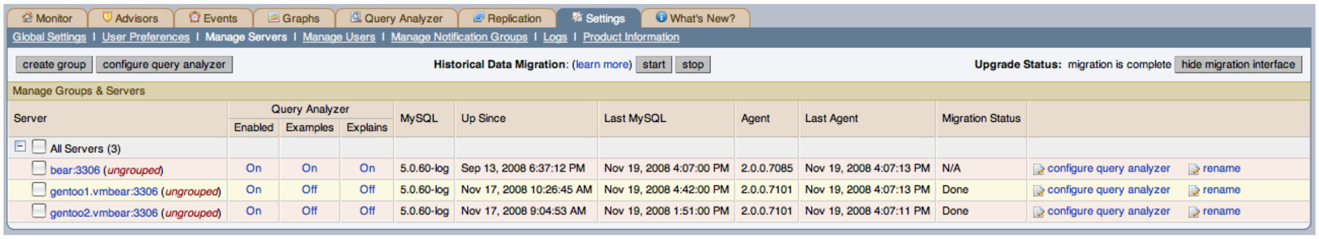
The MySQL Enterprise Service Manager determines the default value for the locale by looking at your browser settings. Changing this value, determines the language setting for any future logins to the Dashboard, overriding your browser settings.

> **Note**
>
> Be sure to set the correct time zone so that alerts are time stamped correctly.
>
> This setting applies only to the specific user.

4.3. Manage Servers

To help with server management, the Service Manager supports the logical grouping of MySQL servers. This allows you to group servers in any fashion you choose. For example, you can manage servers according to purpose. You can group servers by whether the servers handle Internet or intranet data, by whether they power finance or HR applications, or, if you prefer, you may organize them by physical location rather than by functionality.

Figure 4.3. MySQL Enterprise Dashboard: Manage Servers



Historical Data Migration: (learn more)

Upgrade Status: migration is complete

Manage Groups & Servers

Server	Query Analyzer			MySQL	Up Since	Last MySQL	Agent	Last Agent	Migration Status	
	Enabled	Examples	Explains							
⊟ □ All Servers (3)										
□ bear:3306 (*ungrouped*)	On	On	On	5.0.60-log	Sep 13, 2008 6:37:12 PM	Nov 19, 2008 4:07:00 PM	2.0.0.7085	Nov 19, 2008 4:07:13 PM	N/A	configure query analyzer rename
□ gentoo1.vmbear:3306 (*ungrouped*)	On	Off	Off	5.0.60-log	Nov 17, 2008 10:26:45 AM	Nov 19, 2008 4:42:00 PM	2.0.0.7101	Nov 19, 2008 4:07:13 PM	Done	configure query analyzer rename
□ gentoo2.vmbear:3306 (*ungrouped*)	On	Off	Off	5.0.60-log	Nov 17, 2008 9:04:53 AM	Nov 19, 2008 1:51:00 PM	2.0.0.7101	Nov 19, 2008 4:07:11 PM	Done	configure query analyzer rename

For a server to appear in the Dashboard there must be an agent monitoring it. If you wish to add a server to the Dashboard follow the procedure for installing an agent found at Section 2.3, “Monitor Agent Installation”. Instructions for adding a remote server are found at Section 2.3.6.4, “Configuring an Agent to Monitor a Remote MySQL Server”.

The **Manage Servers** panel also allows you control the Query Analyzer and Data Migration. For more information, see Section 8.6, “Query Analyzer Settings” and Section F.9, “Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0”.

> **Note**
>
> The **All Servers** group is built in and every monitored server is a member of this group.

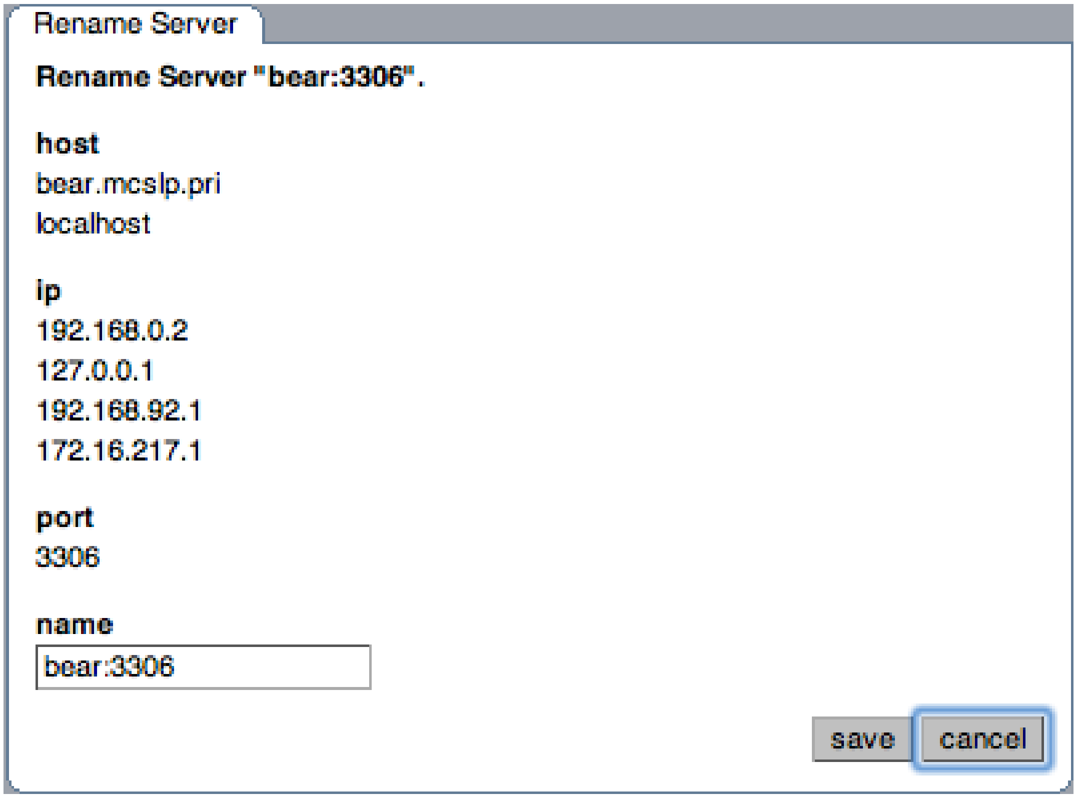
4.3.1. Renaming a Server

You can rename an existing server without losing the current historical data or configuration information. Renaming the server also allows you to modify the name of the server to be more descriptive according to the server's role within your organization. For example, you may want to rename a server from the default host name to include the department and application for the MySQL server.

To rename a server, click the RENAME link next to the server. You will be prompted with information about the server, including the host name and registered IP addresses for the agent. Fill in the alternative name that you want to be displayed in the text box at the bottom of the window.

To rename a server, click context menu (downward arrow) before the current server name, and select **RENAME**. You will be prompted with information about the server, including the host name and registered IP addresses for the agent and the UUID. Fill in the alternative name that you want to be displayed in the text box at the bottom of the window.

Figure 4.4. MySQL Enterprise Dashboard: Server Renaming



Rename Server

Rename Server "bear:3306".

host
bear.mcslp.pri
localhost

ip
192.168.0.2
127.0.0.1
192.168.92.1
172.16.217.1

port
3306

name
bear:3306

> **Note**
>
> Renaming a server using the **MANAGE SERVERS** tab overrides all other settings, including changes made within the agent configuration.

4.3.2. Grouping Servers

All monitored servers are automatically included in the top level server grouping, **ALL SERVERS**. Other server groupings are replication groups or user-defined groups.

You can create a user-defined group by clicking the **MANAGE SERVERS** link. Add a group name and then click the CREATE GROUP button. The new group will be displayed immediately.

Replication groups are automatically discovered by MySQL Enterprise Monitor and in this respect differ from user-defined groups. For more information about replication groups see Chapter 9, *The Replication Page*. However, like user-defined groups you can edit the name of a replication group and add other servers to it.

To add to a group, select the `add to group` link. Choose the server or servers you wish to add and then complete the operation by choosing the ADD TO GROUP button. You can add a server to a group even if the agent is down.

To remove a server from a group expand the server group tree and click the `remove from group` link. To delete a server altogether see Section 4.3.3, “Removing a Server From the Dashboard”.

> **Note**
>
> Slaves removed from a replication group will be rediscovered and re-added to that group.

To delete an group, you can use the **DELETE GROUP** link. This will delete the group and remove the servers from being part of the deleted group.

> **Warning**
>
> You cannot delete automatically created groups, such as those built during replication discovery, unless the group is empty. If the group is not empty, you will get a error.

There are three ways to modify an existing group; by renaming it, adding to it, or removing it. Select the `rename` link to change the name of a group and `add to group` to add servers. Deleting a group simply requires clicking the `remove all from group` link. This removes the server group but has no effect on individual servers.

4.3.3. Removing a Server From the Dashboard

If you no longer wish to monitor a MySQL server you can remove it from the Dashboard. There is no provision for deleting an active server from the Dashboard—to remove a server you must make it inactive by stopping the agent.

For instructions on stopping an agent see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

Once the agent is stopped you may delete the monitored server. Deleting a server simply means that it will no longer show up in the Dashboard.

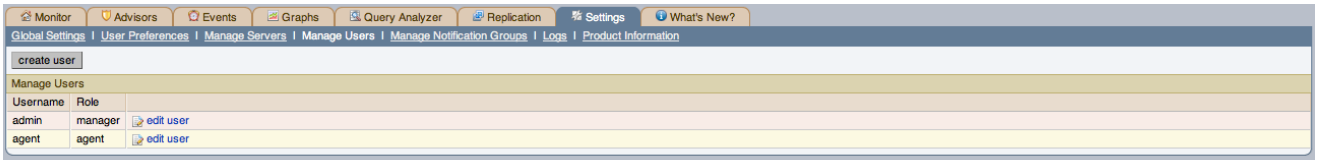
Remove a server by choosing the SETTINGS tab and then the **MANAGE SERVERS** link. Find the server you wish to remove and delete it by clicking the DELETE link. Deleting a server from the **ALL SERVERS** group or from any other group will remove it from the Dashboard entirely.

> **Note**
>
> A DELETE link will not be available for an active server. You must stop the agent before this link will appear.
>
> You may remove a server from any group at any time. Removing the last server from a group also removes that group.

4.4. Managing Users

The **MANAGE SERVERS** panel allows to create, delete and manage individual users that have access to MySQL Enterprise Service Manager

Figure 4.5. MySQL Enterprise Dashboard: Manage Users


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New?
Global Settings I User Preferences I Manage Servers I Manage Users I Manage Notification Groups I Logs I Product Information
create user

Manage Users

Username	Role	
admin	manager	edit user
agent	agent	edit user

To log in to the Dashboard a user account is required. There are three types of users with varying privileges; Administrators, Database Administrators, and Agents. The `Administrator` can create additional users and differs from a `DBA` in this respect. For this reason the `Manage Users` does not display if a DBA user logs in. Additionally, only administrators can change the MySQL Enterprise Credentials section or enter a product key on the `Global Settings` page. These sections do not appear when DBA users log in. For more information on this subject see Section 4.1, “Global Settings”. The `Agent` account simply allows the MySQL Enterprise Monitor Agent to communicate with the Dashboard. There is no need for more than one agent account but defining an account for each server that is monitored can be an advantage since this minimizes exposure should any one agent be compromised. You cannot log in to the Dashboard using the agent's credentials.

When the Dashboard is first launched there are two default users, `Administrator` and `Agent`, both created during installation. Their default user names are respectively, `admin` and `agent`. The Administrator defined during installation as having the root role is unique; this user cannot be deleted.

If you are logged in as an `Administrator`, you can add a new user by choosing the `Manage Users` link from the `Settings` page. To create a user click the CREATE USER button, select a role for the user, and enter a user name and password.

When a new user first logs in, a dialog box opens requesting time zone and locale information. This information may be changed later from the `User Preferences` page. For more information, see Section 4.2, “User Preferences”.

If you installed the Advisors through the Dashboard you should have already configured the settings for the root role user. (See Section 4.1, “Global Settings” and following for more information about this topic.)

> **Warning**
>
> To receive MySQL Enterprise and Advisor updates configure the MySQL Enterprise settings for at least one user. The MySQL Enterprise settings were set up on the first login to the Dashboard. For information on changing these settings see, Section 4.1, “Global Settings”.

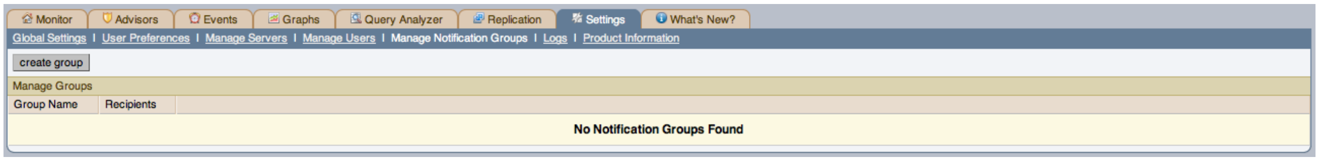
To edit an existing user's information, select the `Manage Users` link, then select the user you wish to edit. Make your desired changes in the fields provided and then save your changes.

To delete an existing user, merely select the `delete` link.

4.5. Manage Notification Groups

The **MANAGE NOTIFICATION GROUPS** panels allows you to create and manage the notification groups used when different notifications and warnings are distributed.

Figure 4.6. MySQL Enterprise Dashboard: Manage Notification Groups


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New?
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information
create group

Manage Groups

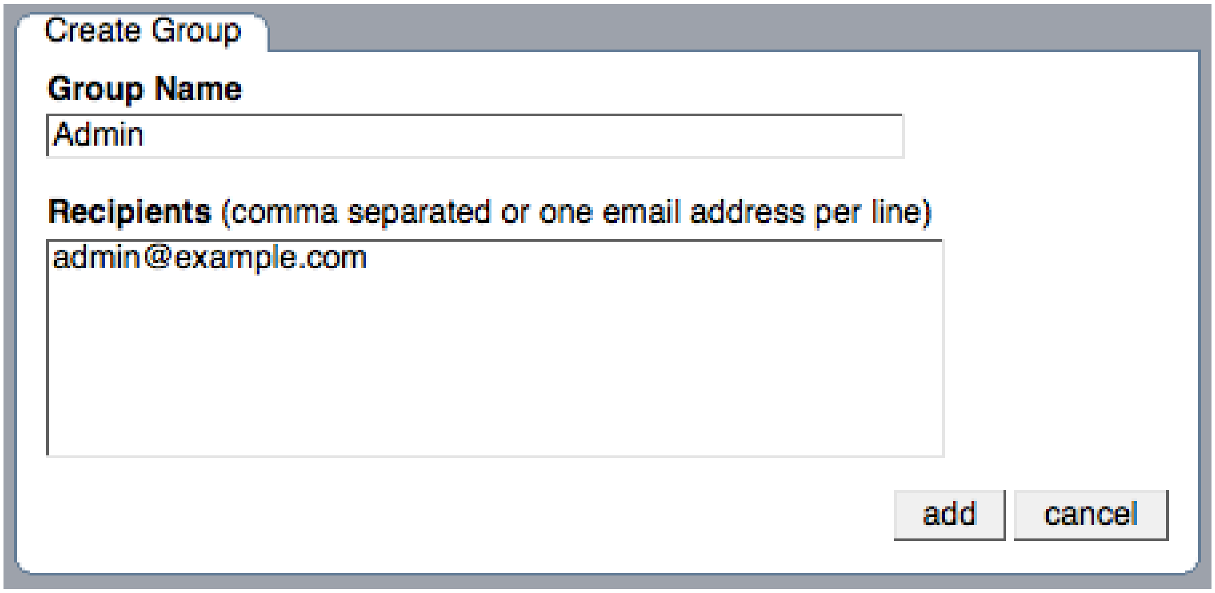
Group Name	Recipients
No Notification Groups Found	

Notification groups are collections of users who should be notified when advisor alerts occur. These users may have login credentials for the Dashboard but this is not a requirement.

You can create a group by clicking the `create group` link. Specify a group name and add recipients. When adding a user an email address must be specified. If you are adding multiple users separate them with commas.

To modify an existing notification group, select the `edit` link next to the group name. Deleting a group simply requires clicking the `delete` link.

Figure 4.7. MySQL Enterprise Dashboard: Edit Notification Groups



Group Name

Admin

Recipients (comma separated or one email address per line)

admin@example.com

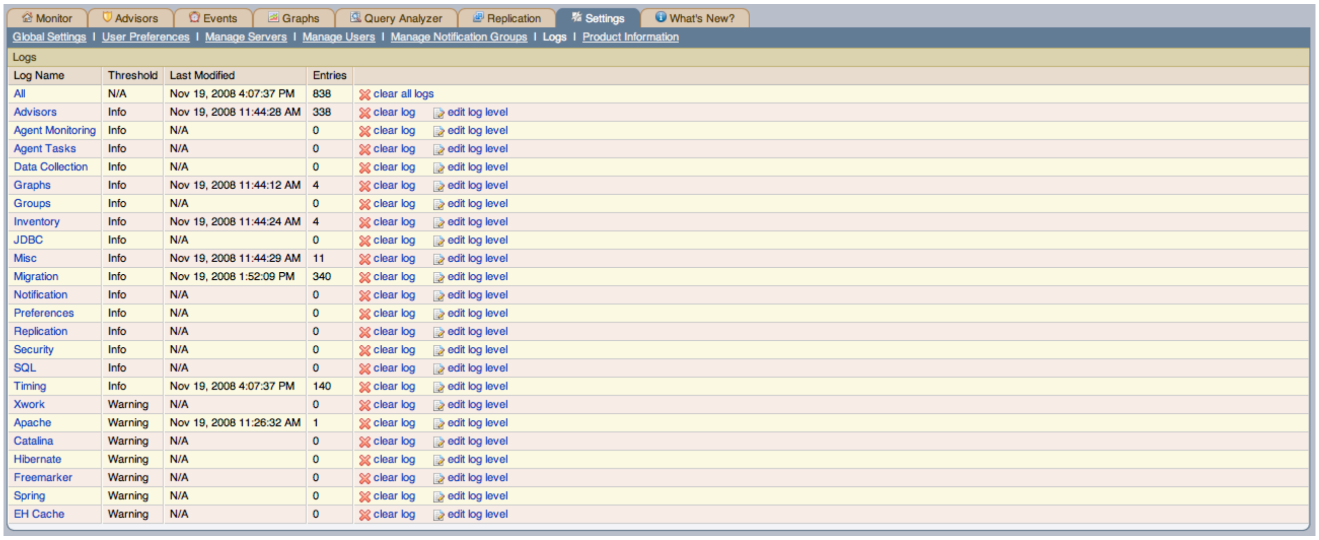
If a rule triggers an alarm, an email will be sent to the members of the notification group specified when the rule was scheduled. For more information about scheduling rules see Section 5.2, "Scheduling Rules".

> **Note**
>
> You should ensure that there is a mail server available for sending out alerts and that there is an account configured for receiving any alerts that are created. For SMS messages, you must ensure that you have configured your SMS service settings.

4.6. Logs

Use the `Logs` link to inspect the various log files associated with the MySQL Enterprise Service Manager. The following image is an example of this screen.

Figure 4.8. MySQL Enterprise Dashboard: Logs


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New?
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information

Logs

Log Name	Threshold	Last Modified	Entries		
All	N/A	Nov 19, 2008 4:07:37 PM	838	☒ clear all logs	
Advisors	Info	Nov 19, 2008 11:44:28 AM	338	☒ clear log	edit log level
Agent Monitoring	Info	N/A	0	☒ clear log	edit log level
Agent Tasks	Info	N/A	0	☒ clear log	edit log level
Data Collection	Info	N/A	0	☒ clear log	edit log level
Graphs	Info	Nov 19, 2008 11:44:12 AM	4	☒ clear log	edit log level
Groups	Info	N/A	0	☒ clear log	edit log level
Inventory	Info	Nov 19, 2008 11:44:24 AM	4	☒ clear log	edit log level
JDBC	Info	N/A	0	☒ clear log	edit log level
Misc	Info	Nov 19, 2008 11:44:29 AM	11	☒ clear log	edit log level
Migration	Info	Nov 19, 2008 1:52:09 PM	340	☒ clear log	edit log level
Notification	Info	N/A	0	☒ clear log	edit log level
Preferences	Info	N/A	0	☒ clear log	edit log level
Replication	Info	N/A	0	☒ clear log	edit log level
Security	Info	N/A	0	☒ clear log	edit log level
SQL	Info	N/A	0	☒ clear log	edit log level
Timing	Info	Nov 19, 2008 4:07:37 PM	140	☒ clear log	edit log level
Xwork	Warning	N/A	0	☒ clear log	edit log level
Apache	Warning	Nov 19, 2008 11:26:32 AM	1	☒ clear log	edit log level
Catalina	Warning	N/A	0	☒ clear log	edit log level
Hibernate	Warning	N/A	0	☒ clear log	edit log level
Freemarker	Warning	N/A	0	☒ clear log	edit log level
Spring	Warning	N/A	0	☒ clear log	edit log level
EH Cache	Warning	N/A	0	☒ clear log	edit log level

The various categories of logs are shown in alphabetic order. The most recent changes to each log are shown in the **LAST MODIFIED** column. The number of entries in any specific log is shown under the **ENTRIES** column.

To view detailed information click the `Log Name`. This will open a separate browser window showing the date, time, alert type, and accompanying message.

On this screen you can filter log information in a couple of ways; by the message type and by time period .

To filter by message type select from the options in the **LEVEL** drop-down box. These are, in order of decreasing severity:

- All
- Error
- Warning
- Information
- Trace
- Debug

You can also adjust the number of items that appear on each page.

Press the `clear all logs` link to remove all log entries. To remove entries of a specific kind click the `clear logs` link associated with the specific log you would like to remove. A confirmation dialog box allows you to back out of this operation and avoid accidentally removing log information.

To clear log files of a specific age see the `Data Purge Behavior` section of the `Global Preferences` page. For more information on this topic see Data Purge Behavior [82].

Use the `edit log level` link to change the type of error logged. The value selected from the `Edit Log Level` dialog box determines what appears under the **THRESHOLD** column (second from the left in Data Purge Behavior [82]).

Selecting `Error` from the list box will create the least number of log entries and `Debug` the most. Choosing `None` turns off logging altogether.

It is also possible to download a compressed version of all the log files. For more information, see Section 4.7, "The Product Information Screen".

4.7. The Product Information Screen

Use the `Product Information` link to view detailed information about your subscription level and contract status.

- **Subscription Warnings**

  The **SUBSCRIPTION WARNINGS** section contains information unique to your subscription. Information provided here will relate to your subscription settings and the servers currently being monitored. For more information, see Section 4.7.1, "Subscription Warning".

- **Contract Status**

  The **CONTRACT STATUS** section displays the subscription level, expiration date, contract number, the number of servers supported, and your MySQL Enterprise user name. The **SUBSCRIPTION LEVEL** section gives more detailed information, including features and any restrictions that may apply. You may update your subscription at any time by clicking the UPDATE button.

- **Subscription Information**

  The **SUBSCRIPTION INFORMATION** section contains detailed information about your subscription level and the numbger of hosts currently monitored within your subscription level.

- **Enterprise Dashboard Server Info**

  The **ENTERPRISE DASHBOARD SERVER INFO** section provides detailed information about the running of your MySQL Enterprise Service Manager, including information about the Java environment, hostname and version information.

The section also includes detailed information about the current status of your MySQL Enterprise Service Manager, showing information on the agents, rules, and outstanding status of the various components. The information provded in this section is listed below:

  - **Pending Heartbeats**
  - **Processed Heartbeats**
  - **Pending Jobs**
  - **Number of Agents**
  - **Monitored mysqld Instances**
  - **Build Version**

- **Diagnostic Report**

  The page includes a hyperlink, **Download diagnostic report**. Click this link to download a compressed version of the MySQL Enterprise Service Manager log files. All the log files found on the `Logs` page (for more information about logs see Section 4.6, “Logs”) are contained in this file. It also contains the Java properties file, the monitored MySQL servers property file, information about the status of the JDBC connection and Java threads, and the `subscription.xml` file. This report is especially useful for debugging the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent.

4.7.1. Subscription Warning

The `Subscriptions Warning` section on the product information page displays any warnings relative to your subscription. For example, if your subscription has expired you may receive a message such as the following:

```
Your Subscription Needs to be Updated

* Your Platinum subscription expired 3 days ago on Feb 14, 2008 11:59:59 PM.

If the subscription information on this page is not current, you can
update it by going to the Enterprise Monitor Global Settings page and
providing MySQL Enterprise credentials or by importing a new product
key that you downloaded from http://www.mysql.com/enterprise/download.php.

To update or renew your subscription, please contact your MySQL
Account Representative at sales@mysql.com or visit
http://www.mysql.com/about/contact/renew.html.  After the update or
renewal is complete you can then follow the above instructions for
updating your subscription.
```

Follow these instructions to update your subscription. If you see this message and your subscription has already been updated, simply click the update button in the **Contract Status** section of this page. This should update your subscription and remove the warning.

> ### Note
>
> After updating your subscription remember to also update your advisors. For instructions on doing this see Section 5.1, “Installing and Updating Advisors”.

Chapter 5. The Advisors Page

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

MySQL Enterprise Advisors are a series of scripts that gather information from your MySQL servers using the Service Manager and the Monitor Agents, analyze that information based on custom rules developed by MySQL AB, and then offer alerts and advice when necessary. As new rules are introduced, the MySQL Enterprise Advisors can be updated through the MySQL Enterprise website.

The MySQL Enterprise Advisors fall into the following categories:

- Administration
    - Better manage databases
    - Suggest improvements for smoother operation
- Heat Chart
    - Drive the status indicators in the Heat Chart
    - Identify up/down status and performance issues
- Performance
    - Identify potential performance bottlenecks
    - Make suggestions for improved database speed
- Replication
    - Identify replication bottlenecks
    - Improve replication design
- Schema
    - Identify schema changes
    - Find security loopholes
- Security
    - Protect MySQL servers
    - Find security loopholes

An advisor category provides a set of rules designed to enforce MySQL best practices for that specific category. Rules can be targeted to run at the individual server or group level and, upon rule violation, provide alerts and expert advice on how to address and correct a problem before it becomes a costly outage.

Individual rules are defined in the `items-mysql-monitor.xml` file. On Windows this file is found in the `C:\Program Files\mySQL\Enterprise\Agent\share\mysql-monitor-agent` directory and on Unix in the `/opt/mysql/enterprise/agent/share/mysql-monitor-agent` directory. Find below the rule for discovering a `root` account with no password.

```
<ITEM>
  <NAME>no_root_password</NAME>
  <FIELD>no_password</FIELD>
  <SCOPE>table</SCOPE>
  <CODE>
    <![CDATA[SELECT COUNT(*) AS no_password FROM mysql.user WHERE user='root' AND password='']]>
  </CODE>
  <NAMESPACE>mysql</NAMESPACE>
  <RETURNS>INTEGER</RETURNS>
  <SOURCE>table</SOURCE>
  <INSTANCE>mysql.user</INSTANCE>
</ITEM>
```

Your MySQL Enterprise subscription level determines which rules are available to you. Subscription levels are cumulative, meaning that higher MySQL Enterprise levels have access to all the rules of the lower levels.

When the Dashboard is first installed, the only rules that are scheduled are those that belong to the `Heat Chart` group.

Go to the Advisors screen by logging in to the Dashboard and choosing the `Advisors` tab.

5.1. Installing and Updating Advisors

Instructions for installing Advisors are given in Section 2.2.7.3, “Installing Advisors After Initial Log-in”, and following. Principally, you need to configure your MySQL Enterprise login or enter your product key before you can update your Advisors.

If your MySQL Enterprise login is configured, you can download the latest Advisors by navigating to the `Advisors` page and finding the `Check for Updates` link. You can periodically update advisors in this way.

> **Note**
>
> If you do not have Internet access and cannot use the online update option you can manually import advisors. This process is described in Section 2.2.7.3, “Installing Advisors After Initial Log-in”.

5.2. Scheduling Rules

Once the MySQL Enterprise Advisors have been installed, you can configure which advisors you would like to run on a scheduled basis.

You can schedule rules by individual server or by group. This is done by first selecting the desired server or server group from the `Server` tree found on the left side of the screen. Next select the `Advisors` tab.

Opening the `Advisors` tab takes you to the `Current Schedule` page. If you have only just installed the MySQL Enterprise Monitor then you will only see the `Heat Chart` group of advisors. Clicking the + button will show all the rules in the Heat Chart group.

Clicking the + button beside any specific rule will show the servers that this rule is running on, its frequency, and its status. Initially, all the `Heat Chart` rules are enabled.

For a more complete description of a rule, click the rule's name. This opens a dialog box that gives detailed information about the rule.

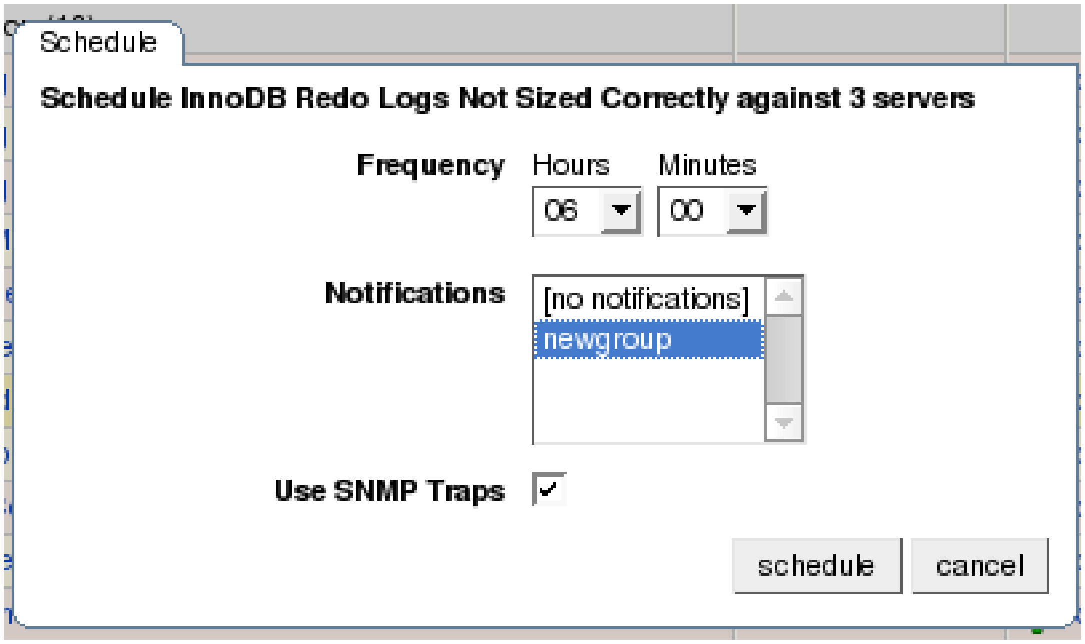
To view the advisors other than the Heat Chart group, select the `Add to Schedule` link. This will show all the advisors available for your subscription level.

Rules are grouped by functionality and displayed in alphabetic order. To expand a group click the + button to the left of the advisor name.

You may activate all the rules in a group by selecting the checkbox beside the group name. Once selected you may apply rules against a specific server or a group of servers. A message showing the group of servers or the specific server you have selected will display immediately below the SCHEDULE button. For example, if the `All Servers` group is selected in the server tree, then the message will read, “Schedule Advisors Against **All Servers**”.

To select a specific rule, expand the group tree by clicking the + button. Select the checkbox to the left of the rule you wish to schedule. Click SCHEDULE to display the following dialog box:

Figure 5.1. MySQL Enterprise Dashboard: Scheduling Dialog



Schedule

Schedule InnoDB Redo Logs Not Sized Correctly against 3 servers

Frequency: Hours 06, Minutes 00

Notifications: [no notifications], newgroup

Use SNMP Traps ☑

The Schedule dialog box allows you to configure the following fields:

- `Frequency`: Dictates how often the rule will run. The default value for different rules varies but a rule can be set to run at any interval desired.

  > **Warning**
  >
  > Setting the frequency of a rule involves tradeoffs. Rule evaluation consumes system resources—CPU, memory, and disk space. While the amount consumed is small, if you run all the rules against dozens of servers on a very frequent basis, you may put a significant load on the Service Manager. So select an appropriate frequency. For example, unless you are stopping and restarting your servers frequently, rules that check server configuration variables probably don't need to run very often.
  >
  > Another consideration is that certain status variables increase monotonically until a server is restarted. Examples of these are `Key_reads`, `Qcache_hits`, `Questions`, `Table_locks_waited`, and similar variables. The value returned by `SHOW STATUS` for these variables is the value since the server was started (or since the last `FLUSH STATUS` command), which is not very useful for performance tuning, especially if the server has been running for an extended period of time. For performance tuning it is much better to know the change in state (for example, delta) of these values over the last 10 minutes, 1 hour, or whatever time frame is appropriate for your application. The frequency at which you schedule a rule is the time frame used to calculate the delta values of these variables, and it is the delta that is used in expression evaluation, not the absolute value. Consequently, select a frequency that is appropriate for the metrics being used in the expression.

- `Notifications`: A listbox of users, notification groups, or both who will be emailed when an advisor reaches an alert level. Single or multiple selections are allowed. For instructions on setting up notification groups see, Section 4.5, "Manage Notification Groups".

Set the frequency, identify whomever you wish to notify, and click SCHEDULE to schedule the advisor. Upon completion, you should see the message, `Successfully scheduled`.

If you haven't set up global SNMP traps and would like your Network Management System (NMS) to handle events related to a specific rule then check the `Use SNMP Traps` checkbox. For more information about Simple Network Management Protocol (SNMP) see Simple Network Management Protocol (SNMP) Traps.

Scheduling rules using the checkbox and the SCHEDULE button is an effective way to schedule multiple rules. To schedule a single rule you may also use the `schedule` link.

When scheduling more than one rule, you have the option of selecting a checkbox to use the default frequency of each rule or you may choose a frequency that will apply to all selected rules. When customizing the frequency, take care that you choose a value that is appropriate to all the rules selected.

> **Note**
>
> If the agent does not have the `SUPER` privilege and InnoDB-related rules are scheduled, a warning will appear in the `DataCollection` log. This also occurs if `mysqld` is started with the `skip-innodb` option. For more information about agent rights see Section 2.3.1, “Creating a MySQL User Account for the Monitor Agent”.

5.2.1. Heat Chart Notifications

It is particularly important that `Notifications` be set for the `Heat Chart` group of rules. This is easily done from the `Current Schedule` page by clicking the + button beside a rule and then clicking a server.

Doing this opens a window with three tabs—`Overview`, `Settings`, and `Advanced`.

The `Overview` tab shows which advisor group a rule belongs to, a description of its purpose, and a link to the history of this alert.

In the `Settings` tab you can adjust the frequency of this rule and also specify a notification group. To select more than one contiguous group press the `Shift` key and click the desired groups. (Some web browsers may require that you drag your selection.) Noncontiguous selections are made by holding down the `Control` key and clicking the desired groups.

If you haven't set up global SNMP traps and would like your Network Management System (NMS) to handle events related to a specific rule then check the `Use SNMP Traps` checkbox. For more information about Simple Network Management Protocol (SNMP) see Simple Network Management Protocol (SNMP) Traps.

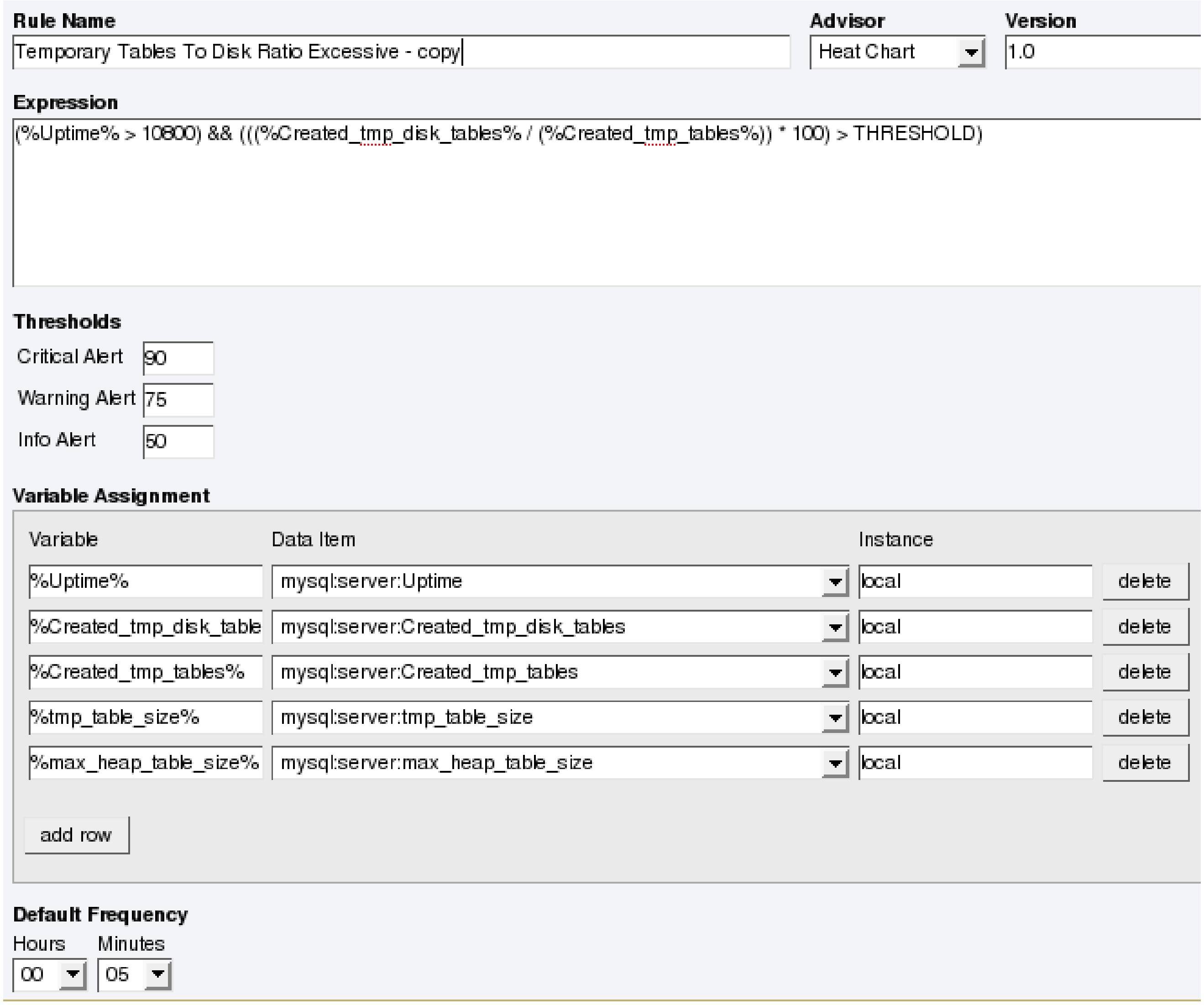
The `Advanced` tab gives detailed information about how this rule is implemented.

5.3. Editing Built-in Rules

The frequency and thresholds defined for a rule are default recommendations. To edit these properties choose the `Create/Edit Rule` link.

The following image shows the screen used to edit rules:

Figure 5.2. MySQL Enterprise Dashboard: Editing Rules



Rule Name

Temporary Tables To Disk Ratio Excessive - copy

Advisor

Heat Chart

Version

1.0

Expression

```
(%Uptime% > 10800) && (((%Created_tmp_disk_tables% / (%Created_tmp_tables%)) * 100) > THRESHOLD)
```

Thresholds

Critical Alert	90
Warning Alert	75
Info Alert	50

Variable Assignment

Variable	Data Item	Instance	
%Uptime%	mysql:server:Uptime	local	delete
%Created_tmp_disk_table	mysql:server:Created_tmp_disk_tables	local	delete
%Created_tmp_tables%	mysql:server:Created_tmp_tables	local	delete
%tmp_table_size%	mysql:server:tmp_table_size	local	delete
%max_heap_table_size%	mysql:server:max_heap_table_size	local	delete

Default Frequency

Hours	Minutes
00	05

Beside the rule name is the `Advisor` drop-down list box, used for setting the advisor group. This list box shows existing groupings and any you may have added. The `Expression` textarea shows the advisor rule, `Variable Assignment` the data item associated with variable(s) used in the rule and `Thresholds` determines when to trigger each alert type.

The three levels of `Thresholds` are `Info Alert`, `Warning Alert`, and `Critical Alert` indicating increasing levels of severity. Levels can be triggered by the expression result being equal to a certain value, greater than a certain value, or less than a certain value.

The data items that variables are associated with are operating system (OS) properties such as available RAM or MySQL characteristics such as the InnoDB buffer pool. To see all available data items drop down the `Data Item` list box. For a listing of these data items see Appendix G, *Data Collection Items*.

In Figure 5.2, “MySQL Enterprise Dashboard: Editing Rules” the drop-down `Data Item` list box within the `Variable Assignment` frame shows the various MySQL server status or operating system specific variables that may be used in expressions. The text boxes below `Thresholds` define the levels at which informational, warning, or critical alerts are issued.

To lower the threshold for an informational alert, simply increase the number given in the `Info Alert` text box.

When a data item can apply to multiple objects, you need to specify which instance to use for that item, hence the `Instance` text box. In almost all cases this should be set to `local`. The exceptions are as follows:

- For CPU-related items set **Instance** to `cpu0`. Additional CPUs on a system are referred to as `cpu1, cpu2` and so on.
- There can be multiple disks mounted on a system. To refer to a specific drive set **Instance** to the name of of the mounted drive. On Windows this would be `C:, D:`, and so on. On Unix systems, use whatever is valid for the `df` command.
- For RAM-related items set **Instance** to `mem`.
- Where there are table-specific variables, the database name and table name must be specified in the **Instance** text box. This topic is discussed in detail in what follows.

> **Note**
>
> It is not possible to have a data item that is unrelated to an instance. This raises the error, `You must map "<variable>" to an instance`, and you will be unable to save the rule.

An agent can only collect data from one MySQL server, so the `instance` entry for a variable in a rule does not need to specify which MySQL server to use; no matter how many servers are being monitored there is always a one-to-one relationship between an agent and its monitored server.

However, on one server there may be multiple occurrences of a variable. For example, there are multiple possible occurrences of table-specific variables such as `Avg_row_length` because there can be multiple databases and tables defined in a MySQL server. In this case, the “instance” refers to the database and table that a data item should be associated with, specified in the form *`databasename`*.*`tablename`*. So, for example, if you want to reference the `Avg_row_length` of the `mysql` database `user` table in an expression, select the `mysql:tablestatus:Avg_row_length` from the **Data Item** list box and specify `mysql.user` in the **Instance** text box.

On the other hand, in the case of a global server variable, there is only one possible target. For example, there can only be one instance of `delay_key_write` because this variable is global and applies to the server as a whole. In this case specify `local` in the **Instance** text box.

To save your changes click the SAVE button at the bottom of the page.

> **Note**
>
> You can change only the thresholds and the frequency of built-in rules. So that rules function properly when updated, other changes are prohibited.
>
> Should you wish to make other changes to a built-in rule, copy it and modify it as desired.

You can edit a rule even if it is currently scheduled. Your changes will not be overwritten when new rules are imported using the `Check for Updates` link.

5.4. Creating Advisors and Rules

In addition to using and editing the advisors and rules provided by MySQL Enterprise, users can create their own advisors and rules to meet their own unique needs. To do this go to the `Advisors` page and choose the `Create/Edit Rule` link.

5.4.1. Creating Advisors

Similar existing rules are grouped together in advisor groups.

The built-in advisors are:

- Administration
- Heat Chart
- Performance
- Replication
- Schema

- Security

The ability to create your own advisor group allows you to create groupings suitable to your circumstances.

You can create your own grouping by simply clicking the CREATE ADVISOR button. Enter an appropriate name and click the ADD button. The newly created group will appear in the `Advisor` column.

The newly created advisor is added to the list box of advisors shown in Figure 5.2, “MySQL Enterprise Dashboard: Editing Rules”. You can now use this category of advisors when you create a new rule.

5.4.2. Overview of Rule Creation

Rules are created using the same screen seen in Figure 5.2, “MySQL Enterprise Dashboard: Editing Rules”. To begin creating a rule from scratch, click the CREATE RULE button. However, the simplest way to create a new rule is to copy an existing one. Unlike editing an existing rule, when you copy a rule, every element of that rule is editable.

You can change the rule name, the advisor group that a rule belongs to and you can set your own version number. In Figure 5.2, “MySQL Enterprise Dashboard: Editing Rules”, you have already seen how the threshold and frequency of a rule may be altered.

Most importantly you can alter a rule's expression. Expressions are the core of a MySQL Enterprise Advisor and are used to define the scenario being monitored. An expression can be as simple as a single server parameter or can be quite complex, combining multiple parameters with various mathematical operations.

An expression has two main characteristics:

- An expression defines a situation where a best practice is not being followed
- The result of an expression must always be 1 or 0 (that is, true or false)

If an expression evaluates to true for a specific server, an alarm is raised, indicating that a best practice is not being followed. If an expression evaluates to false no alarm is raised because the best practice is indeed being followed.

For example, if having binary logging enabled is considered a best practice for a production server (which we believe it is), then this best practice is being violated if `log_bin` is `OFF`. Consequently, the expression for the “Binary Logging Not Enabled” rule is “%log_bin% == OFF”. If this evaluates to 1, an alarm is raised because the best practice is not being followed.

An expression is made up of one or more variables and zero or more mathematical operators. The MySQL Enterprise Monitor uses the MySQL database server's expression parser and evaluator For a complete list of operators and functions see Functions and Operators. For a complete list of the built-in variables used when creating rules see Server Option and Variable Reference.

Creating an expression is dependent on variables defined in the **Variable Assignment** frame. This frame links variables used in the expression field with data gathered from the target MySQL server instance—server status variables, operating system status information, and table information. Variable names are associated with elements in the **Data Item** drop-down list. If you need to define more than one variable simply click the ADD ROW button. For a complete listing of the data collection items used in creating rules see Appendix G, *Data Collection Items*.

The remaining fields determine the information that displays in a notification email or the informational pop-up window associated with each advisor.

> **Note**
>
> When saving a new rule ensure that you do not duplicate the name of an existing rule.

5.4.3. Variables

When an expression is evaluated variables get replaced by values. For example, part of the expression for the “MyISAM Key Cache Has Sub-Optimal Hit Rate” rule calculates the hit rate as follows:

```
100-((%Key_reads% / %Key_read_requests%)*100)
```

If the current value of `%Key_reads%` is 4522 and the current value of `%Key_read_requests%` is 125989, the hit ratio assesses to 96.4%:

```
100 -((4522 / 125989) * 100)
```

By convention, the Advisors supplied by MySQL use '`%`' as the delimiter, for example, `%Key_reads%`. This makes variables more readily identifiable.

In addition to being used in an expression, variables may also be used in the `Description`, `Advice`, `Action`, and `Links` attributes of a rule. This allows you to report the current value of an expression.

For instance, you can add the message, "The current value of Key_reads is %Key_reads%." to the `Advice` text box. When this is displayed on the screen, the value of `%Key_reads%` is substituted into the text. Supposing `%Key_reads%` has a value of `4522`, the message becomes "The current value of Key_reads is 4522."

5.4.4. Thresholds

Each expression has a threshold value that triggers an alert. The `THRESHOLD` keyword is used to associate that value with an alert level—either an `Info`, `Warning`, or `Critical` alert.

For example, the expression for the performance advisor, "Thread Cache Size May Not Be Optimal", is:

```
100-((%Threads_created% / %Connections%) * 100) < THRESHOLD
```

The `THRESHOLD` is set at 95% for an Info level alert, 85% for a Warning alert, and 75% for a Critical alert; producing alerts of three different levels.

Expressions can be quite simple. The expression for "Binary Logging Not Enabled" (one of the Administration alerts) is:

```
%log_bin% == THRESHOLD
```

When the result is `OFF`, only one alert is triggered—a Warning level alert. In this situation you might think we could just use the expression `%log_bin% == "OFF"`. However, doing this would not test binary logging against a threshold so would not result in an alert.

When you create an expression, think carefully about the conditions under which it should be evaluated and the conditions under which it should not. For example, the expression for the "MyISAM Key Cache Has Sub-Optimal Hit Rate" rule is:

```
(%Uptime% > 10800) && (%Key_read_requests% > 10000) »
 && (100-((%Key_reads% / %Key_read_requests%) * 100) < THRESHOLD)
```

The essence of the rule is really: `(100-((%Key_reads% / %Key_read_requests% ) * 100) < THRESHOLD)`. However, when a server is first starting up, it may take a while to reach a state that is representative of normal operations. For example, the key cache and the query cache may need some period of time before they have cached typical application data as opposed to start-up and initialization data. In this case, the first part of the expression, `(%Uptime% > 10800)`, holds off evaluating this expression until the system has been running for 10800 seconds (3 hours).

In addition, if some part of the system is not heavily used an alert may be triggered based on limited data. For example, if your application does not use the MyISAM storage engine, the "MyISAM Key Cache Has Sub-Optimal Hit Rate" rule may be triggered based on very limited use of other MyISAM tables such as the `mysql.user` table. For this reason, this advisor has a second part—`(%Key_read_requests% > 10000)`—meaning the rule won't be evaluated unless there is plenty of activity associated with the key cache.

In other circumstances, there may be periods of time during which you don't want a rule to be evaluated—a blackout period. For example, the expression for the "Slave Too Far Behind Master" rule is: `%Seconds_Behind_Master% > THRESHOLD`. However, suppose you run a backup process between 6 and 7 pm on a replication slave, and it's normal for that slave to get behind the master by an amount more than the THRESHOLD during that time. In that case you don't want to receive an alert because the rule violation is expected. You can achieve this by adding the following to the expression: && CURTIME() NOT BETWEEN '18:00:00' AND '19:00:00' In essence, this means "don't trigger an alert between 18:00:00 and 19:00:00 (6 pm and 7 pm)".

5.4.5. Using Strings

String values may appear in the `Expression` or the `Thresholds` text boxes. In both cases, they must be enclosed within quotation marks. For example, the expression for the "Slave I/O Thread Not Running" rule is:

```
(%Slave_running% == "ON") && (%Slave_IO_Running% != THRESHOLD)
```

In similar fashion the Critical Alerts threshold text box is set to a value of "Yes".

When the expression is evaluated, either "OFF" or "ON" will be substituted for %Slave_running%, and "Yes" or "No" for %Slave_IO_Running%, depending on the state of your system. If the slave is running but the I/O thread is not, the expression then becomes:

```
("ON" == "ON") && ("No" != "Yes")
```

Without quotation marks this expression would not evaluate to TRUE as it should.

> **Note**
>
> So that it is interpreted properly, the == operator is converted to = before being passed to the MySQL expression parser.

5.4.6. Wiki Format

When editing or defining a rule, the text entered in the Problem Description, Advice, Recommended Action, and Links and Further Reading text boxes may be formatted in Wiki format. This allows you to format text and add hyperlinks when creating or editing your own rules.

Find a brief introduction to using Wiki formatting in the following table.

Table 5.1. MySQL Enterprise Monitor: Wiki Formatting

Example	Description
__bold__	boldface text
~~italic~~	italicize text
\\	create a line break
\\ \\	create a double line break
\\\\G	create a backslash
*item 1	create a bulleted list item
#item 1	create a numbered list item
_	use the ‘\’ to escape special characters
{moreInfo:name|url}	create a hyperlink

So the following Wiki text:

```
Replication is a __very nice feature__ of MySQL.  Replication can be very
useful for solving problems in the following areas:
* Data Distribution
* Load Balancing
* Backup and Recovery
You can check replication status and start a slave using the following
commands: SHOW SLAVE STATUS \\\\G\\START SLAVE;
{moreInfo:MySQL Manual: Replication
      FAQ|http://dev.mysql.com/doc/refman/5.0/en/replication-faq.html}
```

Would be translated into the following HTML markup:

```
Replication is a <b>very nice feature</b> of MySQL.  Replication can be very
useful for solving problems in the following areas:
<ul>
  <li>Data distribution</li>
  <li>Load Balancing</li>
  <li>Backup and recovery</li>
</ul>You can check replication status and start a slave with the following
commands: SHOW SLAVE STATUS \G;<br/>START SLAVE;
<a href="http://dev.mysql.com/doc/refman/5.0/en/replication-faq.html"
  target="_blank" >MySQL Manual: Replication FAQ</a>
```

To find out more about this format go to the wikipedia.org web site.

5.4.7. Creating a New Rule: An Example

This section documents the steps required to create a rule. Before attempting to create a rule, please review the preceding sections of this chapter.

This example creates a rule that checks the number of rows in a table. Having 50,000 rows in this table is deemed to warrant a critical alert. Lesser numbers are assigned to informational and warning level alerts.

Begin by navigating to the `Advisors` tab and clicking the `manage rules` link. Then choose the CREATE RULE button.

Create your custom rule by following these steps:

1. Using the `Rule Name` text box, give the rule an appropriate name. Something such as "Excessive number of records in *`table_name`* table", may be appropriate.
2. From the `Advisor` drop down list box choose an advisor group for your rule. The `Administration` group of rules might be suitable but if you wish, create your own group of advisors. For instructions on doing this see Section 5.4.1, “Creating Advisors”.
3. Enter the following expression in the **Expression** text area: '%*`table_name`*_num_rows% > THRESHOLD'. Replace *`table_name`* with the name of the table you wish to monitor. Note that the variable `%table_name_num_rows%` has not yet been defined.
4. Set the Thresholds.
   a. Set the `Critical Alert` level to `50000`.
   b. Set the `Warning Alert` level to `10000`.
   c. Set the `Info Alert` level to `5000`.
5. Define your variable in the `Variable Assignment` frame.
   a. In the `Variable` text box enter `'%table_name_num_rows%`, the variable used in the `Expression` text box
   b. In the `Data Item` drop down list box find and select the `mysql:table:numrows` entry. (For a description of all the data items available see Appendix G, *Data Collection Items*.)
   c. In the `Instance` text box enter *`database_name.table_name`*.
6. Add appropriate entries for the `Problem Description`, the `Advice`, and the `Links` text areas. If you wish, use Wiki markup for these text areas. See Section 5.4.6, “Wiki Format” for more information. Note that you can also reference the `'%table_name_num_rows%` variable in these text areas. For example, you can display the current number of rows with a message such as '*`table_name`* currently has %*`table_name`*_num_rows% rows.'
7. Save the rule.

Once the rule is created it needs to be scheduled against the server that contains the database table you wish to monitor. For instructions on scheduling rules see Section 5.2, “Scheduling Rules”.

5.4.8. Creating a Custom Data Collection Item

Section 5.4.7, “Creating a New Rule: An Example” shows how to create a custom rule and Appendix G, *Data Collection Items* shows the data items that can be used in rule creation. However, in some circumstances you may want to create a rule that uses a custom data collection item.

This section describes how to create a custom data collection item. The steps are as follows:

1. Create an XML file to define how the data is collected.
2. Point the agent configuration file to this XML file.
3. Restart the agent.

5.4.8.1. Creating and Using a Custom Data Item XML File

As an example, this section shows how to create a data item for monitoring the amount of free InnoDB tablespace. The format and content of the XML file that defines the data to be collected is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<classes>
  <class>
    <namespace>mysql</namespace>
    <classname>innodb_min_free</classname>
    <query><![CDATA[SELECT MIN(substring_index(substring_index(table_comment," ",3)," ",-1)/1024/1024)
      as Free FROM INFORMATION_SCHEMA.TABLES WHERE engine = 'InnoDB']]></query>
  </class>
</classes>
```

The above works in MySQL 5.1.23 and earlier. For MySQL 5.1.24 and later, the information is reported within the `data_Free` column, and would need to be declared as:

```
<?xml version="1.0" encoding="utf-8"?>
<classes>
  <class>
    <namespace>mysql</namespace>
    <classname>innodb_min_free</classname>
    <query><![CDATA[SELECT MIN(data_free/1024/1024)
      as Free FROM INFORMATION_SCHEMA.TABLES WHERE engine = 'InnoDB']]></query>
  </class>
</classes>
```

Save this file as:

- Windows: `C:\Program Files\MySQL\Enterprise\Agent\share\mysql-proxy\items\innodb_min_free.xml`
- Unix: `/opt/mysql/enterprise/agent/share/mysql-proxy/items/innodb_min_free.xml`
- Mac OS X: `/Applications/mysql/enterprise/agent/share/mysql-proxy/items/innodb_min_free.xml`

After saving this file, you must point your `mysql-monitor-agent.ini` file to it. (For the location of this file on your operating system see Section 2.3.6.1, “MySQL Enterprise Monitor Agent (`mysql-monitor-agent.ini`) Configuration”.) Find the `[mysql-proxy]` section and add the file name `innodb_min_free.xml` to the `item-files` parameter using a semi-colon as a separator. For example:

```
[mysql-proxy]
 ...
item-files = items-mysql-monitor.xml,innodb_min_free.xml
 ...
```

For this change to take effect you must restart the agent. To do this see:

- Windows: Section 2.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 2.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”

Once the agent has restarted, you will find the new data item in the `Data Item` drop down list box on the `Rule Definition` page. Its fully qualified name is `mysql:table:innodb_min_free`.

5.5. Disabling and Unscheduling Rules

In some circumstances you may no longer wish to apply a rule against a specific server or group of servers and in other circumstances you may want to suspend a rule for a short length of time. With this in mind, it is possible to disable or unschedule a rule.

To disable or unschedule an advisor choose the `Current Schedule` screen of the `Advisors` tab.

Rules may be disabled or unscheduled using the buttons on the upper or lower left of the screen. You may also change a rule by clicking the `enabled` or `unschedule` hyperlink to the right of a rule. The buttons are particularly useful when you are altering more than one rule.

To no longer run a rule against a specific server, expand the advisor group and the specific rule by clicking the + button. You may then click the UNSCHEDULE button. When the dialog window opens, choose the UNSCHEDULE button and that rule will no longer be applied. If you wish to back out of the operation choose CANCEL. If, at a later date, you wish to institute this rule again, you may do so from the `Add to Schedule` page.

If you want to suspend a rule temporarily, use the DISABLE button and follow the same process you would for unscheduling. Once a rule is disabled the link under the status column changes to red and reads `disabled`. When a rule is disabled, data is no longer collected for that rule. A disabled rule is easily re-enabled by clicking the `disabled` link or by using the ENABLE button.

Multiple rules may be altered for one or more servers by selecting the appropriate checkbox and then clicking the UNSCHEDULE, ENABLE, or DISABLE button.

> **Note**
>
> Rules associated with the heat chart cannot be disabled or unscheduled as they are required by MySQL Enterprise Monitor.

5.6. Advisor Blackout Periods

Database servers require regular maintenance and during these periods you may wish to stop Monitor Agents from reporting their findings. During a blackout period rules are not evaluated and notifications are put on hold but Monitor Agents continue to collect data . In this respect blacked-out rules differ from disabled rules; data continues to be collected and stored in the repository.

Blackout periods are enabled by entering the following URL into the address bar of your browser, substituting the appropriate host name, port and server name:

```
http://localhost:18080/rest?command=blackout »
 &server_name=SuSE:3306&blackout_state=true
```

If you are unsure of the host name and port to use, check the `configuration_report.txt` file. Be sure to specify the correct port for the Tomcat server. Specify the server you wish to blackout using the name that appears in the Server Tree, being sure to include a colon and port number as shown in the preceding example.

An HTTP authentication dialog box requesting your Dashboard user name and password will open. Specify the administrator's credentials. The default user name is `admin`; use the password you specified when you initially logged in to the Dashboard.

You can also blackout a server group by entering the following URL into the address bar of your browser, substituting the appropriate host name, and server group name:

```
http://localhost:18080/rest?command=blackout »
 &group_name=Finance&blackout_state=true
```

When the HTTP authentication dialog box opens, enter the administrator's credentials.

You can confirm that a server is blacked out by looking at the server name in the Dashboard; the name of a blacked out server is greyed.

To reactivate the blacked-out server or server group, use the appropriate URL and query string, changing the `blackout_state=true` name/value pair to `blackout_state=false`. Again, this must be done by a user with administrative privileges.

> **Note**
>
> Restarting MySQL Enterprise Monitor will **not** reactivate a blacked out server.

5.6.1. Scripting Blackouts

Rather than opening your web browser and blacking out a server by typing entries into the address bar, you can write a script to achieve the same effect. This section documents a sample blackout script that can be run from the command line.

Create the following file and save it as `blackout.pl`.

```
#!/usr/bin/perl

use LWP 5.64;

# USAGE: blackout.pl servicemanager:18080 admin password servername:3306 true

# $ARGV[0] = management server hostname:port
# $ARGV[1] = management server username
# $ARGV[2] = management server password
# $ARGV[3] = mysqld managed instance server name and port
# $ARGV[4] = blackout state (true/false)


my $browser = LWP::UserAgent->new;
$browser->credentials(
  $ARGV[0],
  '',
  $ARGV[1],
  $ARGV[2]
);

my $url = URI->new('http://'.$ARGV[0].'/rest');

$url->query_form( # And here the form data pairs:
  'command' => 'blackout',
  'server_name' => $ARGV[3],
  'blackout_state' => $ARGV[4]
);

my $response = $browser->post( $url );

if (!$response->is_success) {
  die $response->status_line . "\n";
}
```

> **Note**
>
> Windows users can omit the shebang line.

On Unix systems use the `chmod +x blackout.pl` command to make the file executable.

At the command line enter **`blackout.pl`** ***`servicemanager:18080`*** **`admin`** ***`password servername:3306`*** **`true`**.

If you are unsure of the host name and port to use, check the `configuration_report.txt` file. Be sure to specify the correct port for the Tomcat server. Specify the server you wish to blackout using the name that appears in the Server Tree, being sure to include a colon and port number as shown in the preceding example. Make sure that the user you specify is a "manager". Specifying a user with "dba" rights only will not black out a server and no error will be displayed.

You can confirm that a server is blacked out by looking at the server name in the Dashboard; the name of a blacked out server is greyed. To end the blackout, run the same script, changing the final argument to `false`.

> **Note**
>
> Restarting MySQL Enterprise Monitor will **not** reactivate a blacked out server.

Chapter 6. The Events Page

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

Once an advisor has been scheduled, it will run at set intervals. If it finds nothing of interest no alerts or emails will be created.

Events are defined using a number of different levels according to the severity of the alert. The seven levels are

- **Failure**: Indicates a failure for the given rule or event. Usually this indicates that the rule on which the event is based has failed, probably due to a typo or other error. Occasionally it can indicate a failure to retrieve the required information for a given rule.
- **Critical**: The event is critical and requires immediate attention. Critical events normally indicate that a serious problem has, or is about, to occur. You should examine the event and determine the cause and fix it as soon as possible.
- **Warning**: The event is a warning of something that you should be aware of, but which is not affecting the operation of your server.
- **Info**: The event is for informational purposes. Info events are used to indicate issues that do not affect the operation of your server, but which may highlight a minor configuration or other issue.
- **Success**: The rule has executed successfully with no issues. This is used to indicate that the result of the rule is OK and is used to indicate when an event that was previously in a Critical or Failure state has returned to normal.
- **Unknown**: Indicates when the current state of the event/rule is currently unknown. This state can be issued when the current status of the event cannot be determined.
- **Closed**: The issue has been corrected and marked closed.

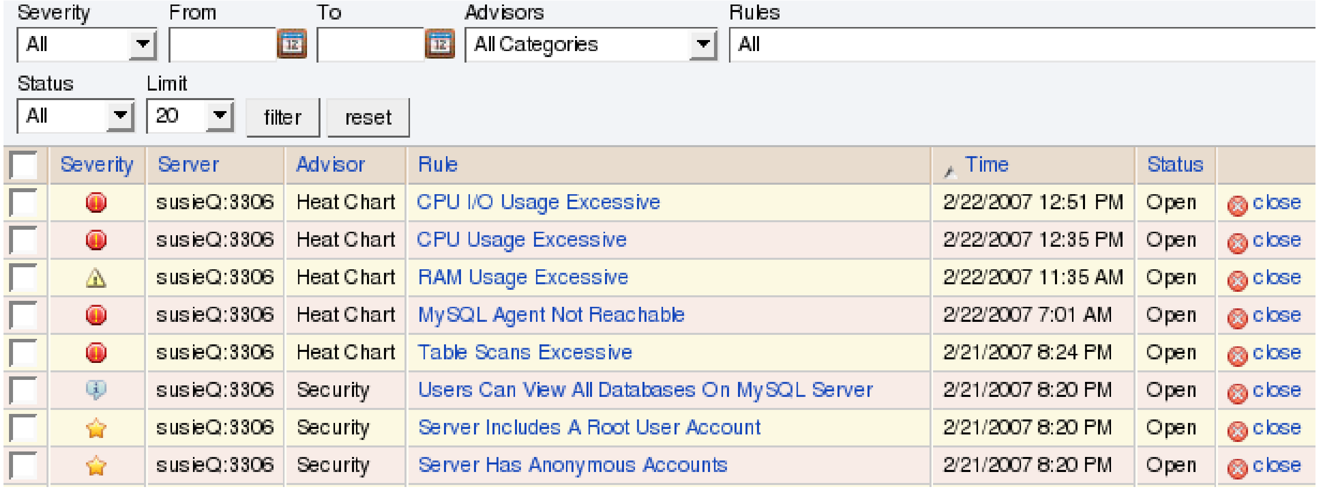
For convenience, the event states are also represented by icons, as defined in the table below. The table also shows the relative level of the alert from the highest (Critical) to the lowest (unknown). The order represented here is used when identifying when an event has transitioned between levels (for example, from Success to Critical), hence triggering a notification, and when sorting events based on their current state.

Icon	Description
❗	An octagonal red icon indicates a critical alert.
⚠	A triangular yellow icon a warning.
💬	A conversation bubble an informational alert.
★	A star beside an event indicates that the rule has run successfully and no alert created.
❓	A question mark icon indicates that the status of the rule is unknown.

When alerts are triggered, they appear on the **Events** screen. Alerts also appear on the `Monitor` screen in order of severity. The notification group or groups associated with a specific rule receive email notification when an alert is triggered. For more information about creating notification groups see Section 4.5, “Manage Notification Groups”.

To view open events, click the **Events** tab. The tree-view on the left determines which server or server group these events belong to. Open events are shown in tabular format.

Figure 6.1. MySQL Enterprise Dashboard: Events Screen



Severity: All
From:
To:
Advisors: All Categories
Rules: All
Status: All
Limit: 20

□	Severity	Server	Advisor	Rule	Time	Status	
□	●	susieQ:3306	Heat Chart	CPU I/O Usage Excessive	2/22/2007 12:51 PM	Open	⊗ close
□	●	susieQ:3306	Heat Chart	CPU Usage Excessive	2/22/2007 12:35 PM	Open	⊗ close
□	⚠	susieQ:3306	Heat Chart	RAM Usage Excessive	2/22/2007 11:35 AM	Open	⊗ close
□	●	susieQ:3306	Heat Chart	MySQL Agent Not Reachable	2/22/2007 7:01 AM	Open	⊗ close
□	●	susieQ:3306	Heat Chart	Table Scans Excessive	2/21/2007 8:24 PM	Open	⊗ close
□	ℹ	susieQ:3306	Security	Users Can View All Databases On MySQL Server	2/21/2007 8:20 PM	Open	⊗ close
□	★	susieQ:3306	Security	Server Includes A Root User Account	2/21/2007 8:20 PM	Open	⊗ close
□	★	susieQ:3306	Security	Server Has Anonymous Accounts	2/21/2007 8:20 PM	Open	⊗ close

The event table has the following columns:

- **Severity**: An icon indicating the severity of the alert
- **Server**: The name of the server the alert applies to
- **Advisor**: The category of the advisor
- **Rule**: A short description of the rule that has been violated
- **Time**: The approximate time the event occurred
- **Status**: The status of the event
- **Unnamed Column**: Provides a link to the `Close` dialog box

By default, all events are shown but the list of events can be filtered using the form displayed above the event list. The options include filtering by:

- Severity
- Date (using a range with From/To)
- Advisor group
- Specific rule
- Status

Choose the options you are interested in and click the filter button to refresh the display. You may limit the number of items that appear on a page by choosing a different value from the **Limit** drop down listbox.

The drop down list box showing severity has the options: `All`, `Alerts`, `Critical`, `Warning`, `Info`, `Success`, and `Unknown`. Selecting the option `All` shows all alerts and also those rules that have run successfully.

A successful rule is one that has not been violated and is indicated by a star icon.

The `Alerts` shows only those rules that have been violated.

Columns can be sorted by clicking the individual column headings. The alerts shown in Figure 6.1, “MySQL Enterprise Dashboard: Events Screen”.

The server shown in Figure 6.1, “MySQL Enterprise Dashboard: Events Screen”, is filtered for `All`. Typically, when filtering by severity you would choose `Alerts` and, if you see a `Critical`, `Warning`, or `Info` alert, use the `All` filter to see when the rule last ran successfully. This may assist in determining the cause of the alert.

Besides filtering for severity, you can also choose to filter for a specific time period using the `From` and `To` text boxes. You also have the choice of filtering by specific rules or categories of rules. The `Status` drop-down list box let's you choose `All`, `Open`, or `Closed` events. To avoid excessive scrolling, you can also limit the number of events that show on a specific page.

For more information about an alert, click the rule name. A pop-up window will appear showing a description of the alert and the exact time of occurrence. This pop-up windows provides links to useful resources and advice for resolution. You can also view the exact expression that generated the event.

6.1. Closing an Event

After determining what action to take, events should be closed.

To resolve an individual alert click the `close` link in the **OPERATIONS/NOTES** column. Document the resolution using the `Notes` text area and choose the CLOSE EVENT(S) button.

To close a number of alerts simultaneously, select the checkbox beside the alerts you wish to close and then click the CLOSE button to the lower or upper left side of the screen.

Once an event has been closed it appears on the **EVENTS** screen showing a `resolution notes` link. Click this link to review the notes. Events that have been closed are saved in the Repository. If you wish to view closed events filter the display by choosing `Closed` from the **STATUS** drop-down box.

6.2. Notification of Events

If you have created notification groups and then configured individual rules to send their notification to one or more groups then the notification works as follows:

- Notification of an event takes place when a rule is executed and the severity level for a given rule changes. For example, if during execution of a rule, the result of the rule goes from severity level `Success` to `Critical`, the configured notification groups will get one email with the detail of the event and current status.
- If the status of the event has not changed when the rule is next executed, no new email is sent.

  This is true whether the rule is executed as part of it's normal schedule or execution of the rule has been explicitly requested.

SNMP notifications are sent each time a given rule is executed, irrespective of the current or returned state.

Chapter 7. The Graphs Page

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

Navigate to the `Graphs` page by choosing the **GRAPHS** tab.

By default four graphs are displayed on the `Monitor page`. These graphs present information about the currently selected server or server group, showing the hit ratios, CPU utilization, connections, and database activity. Color coding helps distinguish different aspects of each graph.

From the `Monitor` page you can make permanent or temporary changes to the way a graph is displayed. For example, you can choose to display the last hour's activity or you can choose to view a specific period of time.

Persistent changes to the way the graphs display are only made from the `Monitor` page. You can set the size of the thumbnails and the full-sized graphs and you can also set their refresh interval. For more information, see Section 3.2, “The Server Graphs and Critical Events”. As with the `Monitor` page, the data shown in the graphs is determined by the server or group of servers selected in the server tree.

The `Graphs` page shows all the available graphs and provides the capability of adjusting the scale of the graphs, allowing a more or less detailed view as the situation requires. To ensure that you have the latest versions of the various graphs click on the **CHECK FOR UPDATES** link on the top left of this page.

7.1. Displaying Graphs

The total number of graphs varies depending upon your subscription level. The four graphs that appear by default on the `Monitor` page are:

- Hit Ratios
- Database Activity
- Connections
- CPU Utilization

When the `Graphs` page is first opened, no graphs are visible. To view a graph click the + button on the left or, to view all graphs, use the EXPAND ALL button.

The larger size of graphs is the primary reason for viewing graphs on the `Graphs` page rather than on the `Monitor` page. Additionally, you can only show a maximum of six graphs on the `Monitor` page; the remaining graphs can only be viewed from the `Graphs` page.

7.2. Setting an Interval

Change the interval for a graph by choosing values from the **HOURS** and **MINUTES** drop-down list boxes. If necessary adjust the width and height of the graph and then click the UPDATE button. The changes to the time span apply to all the graphs on the `Graphs` page but have *no* effect on the graphs on the `Monitor` page.

To change the graphs both here and on the `Monitor` page use the `configure graphs` link on the top right. This opens a dialog box for setting the default interval for the x-axis. Save any changes that you have made and the values chosen will be the defaults whenever you log in. You can also change the defaults from the `Monitor` page as described in Section 3.2, “The Server Graphs and Critical Events”; defaults for other users will be unchanged.

Use the RESET button to restore the default value for the interval. Doing this will also reset the default size of the graphs.

7.3. Setting a Time Span

Setting a graph to display a time span gives you a historical perspective on server activity. You may want to know what was happening at a specific point in time or you may wish to look at an extended period to determine patterns or trends. Changing the time span gives you the flexibility to do this.

In the **TIME DISPLAY** drop-down list box select the `From/To` option. Choosing this option updates the display to include **TO** and **FROM** text boxes.

Set the date you wish to start viewing from by manually entering the date in year, month, and day format (2007-03-14). However, it is much easier to click the calendar icon and choose a date from the drop-down calendar. Enter a terminating date in the same way. If you wish, you may also choose the specific time of day by selecting the hour and minute.

If necessary adjust the width and height of the graph and then click the UPDATE button. The changes to the time span apply to all the graphs on the `Graphs` page but have *no* effect on the graphs on the `Monitor` page. You cannot change the time span of the graphs that appear on the `Monitor` page. Changes apply only to the current user; defaults for other users will be unchanged.

Use the RESET button to cancel your changes.

Chapter 8. The Query Analyzer Page

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

Query Analyzer enables you to monitor the statements being executed on a monitored server and retrieve information about the query, number of executions and the execution times of each query. Queries are normalized, so that the unique data defined within each query has been removed. By removing the data specific elements of the queries, the generic queries can be counted and identified more easily.

> **Important**
>
> MySQL Query Analyzer is designed to gather query performance information from a variety of sources. Query Analyzer uses a new agent plug-in to proxy your queries and collect performance data that is then transmitted to the Enterprise Monitor. This is a new role for the Agent: it is no longer just monitoring, it is now *optionally* between your MySQL client application and the mysql server.
>
> Depending upon your system load, it is possible to overload the proxy or have the proxy/agent consume system resources needed by mysql itself. In particular, the memory needed by the MySQL Agent for basic monitoring is fairly small and consistent and depends on the number of rules you have enabled. However, when the Query Analyzer is enabled, the Agent can use significantly more memory to monitor and analyze whatever queries you direct through it. In this case, the amount of memory used depends on the number of unique normalized queries, example queries and example `EXPLAIN`s being processed plus the network bandwidth required to send this query performance data to the Service Manager. In general, the amount of memory used for the Query Analyzer is well-bounded, but under heavy load or, in some cases under older versions of linux, RAM usage by Query Analyzer may be too high for your environment and load.
>
> Therefore we advise you to use this release of Query Analyzer extensively in development, test and stage environments under load for an extended period of time before considering usage in a production environment. For all deployments:
>
> 1. Carefully monitor the Agent's resource consumption using the new graph **Memory Usage - Agent** graphs available on the **Graph** tab. You can also add an SMTP or SNMP notification to the new Heat Chart rule **MySQL Agent Memory Usage Excessive**.
>
> 2. If the amount of memory consumed is too high, consider sampling queries during nonpeak hours or monitoring only a subset of queries on this system.
>
> If you experience any problems with Query Analyzer, we're interested in working with you closely and quickly to resolve them. Please open a Support issue right away. We're already working hard on optimizing Agent/proxy RAM usage and are planning a series of rapid releases to quickly distribute these and other improvements to you.

Query Analyzer works by intercepting the SQL statements that your MySQL client application sends to the MySQL server. Instead of connecting direct to the MySQL Server, queries are routed through the MySQL Enterprise Monitor Agent, the agent/proxy forwards the queries on to the server and sends the replies back to the client application as normal. In addition to forwarding the queries, the agent/proxy will also normalize the queries and then supply the execution information about each query to the monitor. The forwarding functionality is provided by the same module that supports the MySQL Proxy application. For information on MySQL Proxy, see MySQL Proxy.

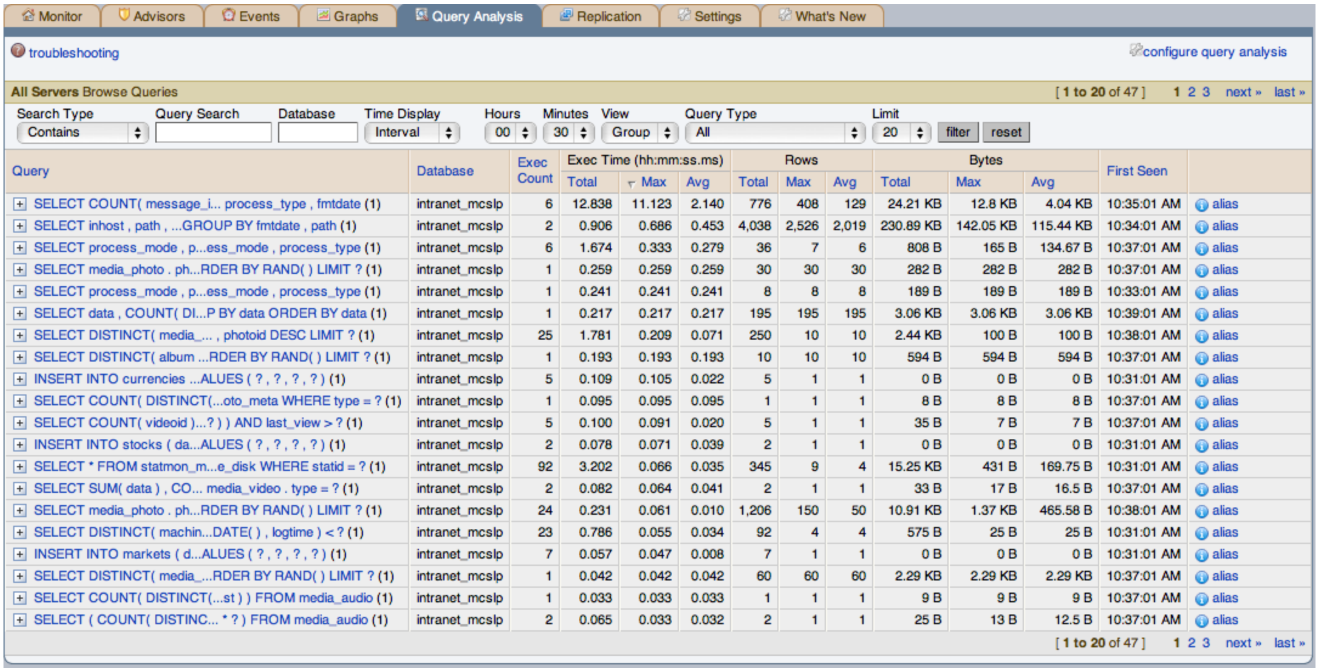
> **Important**
>
> The MySQL Proxy component, and Query Analyzer, require that clients connecting through MySQL Enterprise Monitor Agent are using MySQL 5.0 or later. Clients that use the library provided with MySQL 4.1 or earlier will not work with MySQL Enterprise Monitor Agent.

Once your MySQL client application has been configured to communicate using the MySQL Enterprise Monitor Agent, queries will be monitored and the simplified queries, without the query specific data, will be sent to the MySQL Enterprise Monitor Agent.

There are a number of different ways that you can enable Query Analysis. For more information on the different options, see Section 8.1, “Enabling Query Analyzer”.

To analyse the queries captured by the agent/proxy, change to the `Query Analyzer` page. You can see an example of the table on that page in the figure below.

Figure 8.1. MySQL Enterprise Dashboard: Query Analyzer



All Servers Browse Queries — [**1 to 20** of 47] **1** 2 3 next » last »

Search Type: Contains | Query Search: | Database: | Time Display: Interval | Hours: 00 | Minutes: 30 | View: Group | Query Type: All | Limit: 20

Query	Database	Exec Count	Exec Time (hh:mm:ss.ms)			Rows			Bytes			First Seen	
			Total	Max	Avg	Total	Max	Avg	Total	Max	Avg		
⊞ SELECT COUNT(message_i... process_type , fmtdate (1)	intranet_mcslp	6	12.838	11.123	2.140	776	408	129	24.21 KB	12.8 KB	4.04 KB	10:35:01 AM	alias
⊞ SELECT inhost , path , ...GROUP BY fmtdate , path (1)	intranet_mcslp	2	0.906	0.686	0.453	4,038	2,526	2,019	230.89 KB	142.05 KB	115.44 KB	10:34:01 AM	alias
⊞ SELECT process_mode , p...ess_mode , process_type (1)	intranet_mcslp	6	1.674	0.333	0.279	36	7	6	808 B	165 B	134.67 B	10:37:01 AM	alias
⊞ SELECT media_photo . ph...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	1	0.259	0.259	0.259	30	30	30	282 B	282 B	282 B	10:37:01 AM	alias
⊞ SELECT process_mode , p...ess_mode , process_type (1)	intranet_mcslp	1	0.241	0.241	0.241	8	8	8	189 B	189 B	189 B	10:33:01 AM	alias
⊞ SELECT data , COUNT(DI...P BY data ORDER BY data (1)	intranet_mcslp	1	0.217	0.217	0.217	195	195	195	3.06 KB	3.06 KB	3.06 KB	10:39:01 AM	alias
⊞ SELECT DISTINCT(media_... , photoid DESC LIMIT ? (1)	intranet_mcslp	25	1.781	0.209	0.071	250	10	10	2.44 KB	100 B	100 B	10:38:01 AM	alias
⊞ SELECT DISTINCT(album ...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	1	0.193	0.193	0.193	10	10	10	594 B	594 B	594 B	10:37:01 AM	alias
⊞ INSERT INTO currencies ...ALUES (? , ? , ? , ?) (1)	intranet_mcslp	5	0.109	0.105	0.022	5	1	1	0 B	0 B	0 B	10:31:01 AM	alias
⊞ SELECT COUNT(DISTINCT(...oto_meta WHERE type = ? (1)	intranet_mcslp	1	0.095	0.095	0.095	1	1	1	8 B	8 B	8 B	10:37:01 AM	alias
⊞ SELECT COUNT(videoid)...?)) AND last_view > ? (1)	intranet_mcslp	5	0.100	0.091	0.020	5	1	1	35 B	7 B	7 B	10:37:01 AM	alias
⊞ INSERT INTO stocks (da...ALUES (? , ? , ? , ?) (1)	intranet_mcslp	2	0.078	0.071	0.039	2	1	1	0 B	0 B	0 B	10:31:01 AM	alias
⊞ SELECT * FROM statmon_m...e_disk WHERE statid = ? (1)	intranet_mcslp	92	3.202	0.066	0.035	345	9	4	15.25 KB	431 B	169.75 B	10:31:01 AM	alias
⊞ SELECT SUM(data) , CO... media_video . type = ? (1)	intranet_mcslp	2	0.082	0.064	0.041	2	1	1	33 B	17 B	16.5 B	10:37:01 AM	alias
⊞ SELECT media_photo . ph...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	24	0.231	0.061	0.010	1,206	150	50	10.91 KB	1.37 KB	465.58 B	10:38:01 AM	alias
⊞ SELECT DISTINCT(machin...DATE( ) , logtime) < ? (1)	intranet_mcslp	23	0.786	0.055	0.034	92	4	4	575 B	25 B	25 B	10:31:01 AM	alias
⊞ INSERT INTO markets (d...ALUES (? , ? , ? , ?) (1)	intranet_mcslp	7	0.057	0.047	0.008	7	1	1	0 B	0 B	0 B	10:31:01 AM	alias
⊞ SELECT DISTINCT(media_...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	1	0.042	0.042	0.042	60	60	60	2.29 KB	2.29 KB	2.29 KB	10:37:01 AM	alias
⊞ SELECT COUNT(DISTINCT(...st)) FROM media_audio (1)	intranet_mcslp	1	0.033	0.033	0.033	1	1	1	9 B	9 B	9 B	10:37:01 AM	alias
⊞ SELECT (COUNT(DISTINC... * ?) FROM media_audio (1)	intranet_mcslp	2	0.065	0.033	0.032	2	1	1	25 B	13 B	12.5 B	10:37:01 AM	alias

[**1 to 20** of 47] **1** 2 3 next » last »

The main Query Analyzer table provides the summary information for all of the queries executed using the agent/proxy. The table will track all the queries submitted to the server using the agent/proxy. The table will show a maximum of 20 rows, and you can page through the list of queries by using the page numbers, or the **NEXT**, **PREVIOUS**, **FIRST**, and **LAST** buttons. To filter the list of queries that are displayed, or to change the number of queries, see Section 8.3, “Filtering Query Analyzer Data”.

Each row within the table provides the statistical information for one normalized query statement. If you have configured multiple agent/proxies to accept and forward queries to different servers, then you can expand the server view. The summary information displayed is different depending on whether you have selected a server group or an individual server.

If you have selected a server group, then the information displayed is aggregated from across the entire group. The same query executed on multiple servers will show average, total and minimum/maximum information for that query across all the servers. If you select an individual server, then only queries executed on that server are included within the table.

For each row, the following columns are populated according to the selected filtering options. For example, if the filter have been configured to show queries within the last 30 minutes (**Interval**), then only queries executed during that time will be displayed, and the corresponding statistics, such as execution times, rows returned and bytes returned will be according to that 30 minute timespan.

- **Query**: The normalized version of the query. Normalization removes the query-specific data so that different queries with different data parameters are identified as the same basic query.

  The information is shown as one query per row. Each query row is expandable, and can be expanded to show the execution times for individual servers for that query.

- **Database**: The default database in use at the time of the query. The database name may not match the database used within the query if you have have explicitly stated the database name in the query.

- **Exec Count**: The number of times that the query has been executed.

- **Exec Time**: The execution time for all the matching queries. This is the time, for every invocation of the corresponding query, as calculated by comparing the time when the query was submitted and when the results were returned by the server. Times are expressed in HH:MM:SS.MS (hours, minutes, seconds, and milliseconds).

  The **Execution** column is further subdivided into the following columns:

  - **COUNT**: The total number of executions.
  - **TOTAL**: The cumulative execution time for all the executions of this query.
  - **MAX**: The maximum execution time for an execution of this query.
  - **AVG**: The average execution time for the execution of this query.

  When looking at the information provided in this query, you should consider comparing the average and maximum execution times to see if there was a problem on a specific server or during a specific time period when the query took place, as this may indicate an issue that needs to be investigated. For more information, see Section 8.4, “Using Query Analyzer Data”.

  > **Note**
  >
  > Due to limitations in the counters used for monitor queries on Windows, the time reported for short queries may be reported 0.000. This will be fixed in a future release.

- **ROWS**: The rows returned by the query. The column is sub-divided into the following columns:
  - **TOTAL**: The sum total number of rows returned by all executions of the query.
  - **MAX**: The maximum number of rows returned by a single execution of the query.
  - **AVG**: The average number of rows returned by all executions of the query.
- **BYTES**: The number of bytes returned by each query. The column is sub-divided into the following columns:
  - **TOTAL**: The sum total bytes returned by all executions of the query.
  - **MAX**: The maximum number of bytes returned by a single execution of the query.
  - **AVG**: The average number of bytes returned by all executions of the query.
- **FIRST SEEN**: The first time the query was seen within the given filter conditions.

You can sort the list of queries by clicking the column name. The direction of the sort (highest to lowest, or lowest to highest) is indicated by a triangle next to the currently selected column. The default is to sort the list of queries by the Total Execution time.

8.1. Enabling Query Analyzer

There are three different ways of enabling query analyzer:

- Change your MySQL client application to talk to the Proxy port you configured during installation. This requires changing your MySQL client application code, and may require that you stop and restart your MySQL client application , but does not require any changes to your MySQL server. For more information, see Section 8.1.1, “Enabling Query Analyzer by Changing the MySQL Client Application”.
- Change your MySQL server to listen on a different port, and configure the Agent/proxy to listen on the original MySQL server port. This does not require any changes to your MySQL client application , but will require shutting down and restarting your MySQL server, which may affect your cache and performance. For more information, see Section 8.1.2, “Enabling Query Analyzer by Changing MySQL Server”.
- Use IP tables to redirect the network packets to the agent/proxy.

> **Important**
>
> MySQL Query Analyzer is designed to gather query performance information from a variety of sources. Query Analyzer uses a new agent plug-in to proxy your queries and collect performance data that is then transmitted to the Enterprise Monitor. This is a new role for the Agent: it is no longer just monitoring, it is now *optionally* between your MySQL client application and the mysql server.

> Depending upon your system load, it is possible to overload the proxy or have the proxy/agent consume system resources needed by mysql itself. In particular, the memory needed by the MySQL Agent for basic monitoring is fairly small and consistent and depends on the number of rules you have enabled. However, when the Query Analyzer is enabled, the Agent can use significantly more memory to monitor and analyze whatever queries you direct through it. In this case, the amount of memory used depends on the number of unique normalized queries, example queries and example `EXPLAIN`s being processed plus the network bandwidth required to send this query performance data to the Service Manager. In general, the amount of memory used for the Query Analyzer is well-bounded, but under heavy load or, in some cases under older versions of linux, RAM usage by Query Analyzer may be too high for your environment and load.
>
> Therefore we advise you to use this release of Query Analyzer extensively in development, test and stage environments under load for an extended period of time before considering usage in a production environment. For all deployments:
>
> 1. Carefully monitor the Agent's resource consumption using the new graph **Memory Usage - Agent** graphs available on the **Graph** tab. You can also add an SMTP or SNMP notification to the new Heat Chart rule **MySQL Agent Memory Usage Excessive**.
> 2. If the amount of memory consumed is too high, consider sampling queries during nonpeak hours or monitoring only a subset of queries on this system.
>
> If you experience any problems with Query Analyzer, we're interested in working with you closely and quickly to resolve them. Please open a Support issue right away. We're already working hard on optimizing Agent/proxy RAM usage and are planning a series of rapid releases to quickly distribute these and other improvements to you.

Note that you must have enabled Query Analyzer within the agent/proxy during installation. If you did not enable Query Analyzer during the installation of the agent/proxy, check the following elements within the main `mysql-monitor-agent.ini` configuration file:

- Add the `proxy` plugin to the `plugins` parameter:

  ```
  plugins=proxy,agent
  ```

- Ensure that the `quan.lua` items file is enabled in the `agent-item-files` configuration property:

  ```
  agent-item-files = share/mysql-proxy/items/quan.lua,share/mysql-proxy/items/items-mysql-monitor.xml
  ```

- Check and set the `proxy-address`, `proxy-backend-addresses`, and `proxy-lua-script` settings are configured:

  ```
  proxy-address=:4040
  proxy-backend-addresses = 127.0.0.1:3306
  proxy-lua-script        = share/mysql-proxy/quan.lua
  ```

For more information on these configuration options, see Section 2.3.6.1, “MySQL Enterprise Monitor Agent (`mysql-monitor-agent.ini`) Configuration”.

> **Note**
>
> The Query Analyzer functionality may show as being enabled on a server, even though the modules within MySQL Enterprise Monitor Agent may not have been enabled.

You may also need to make some additional changes to the security configuration on your server to ensure that queries are correctly reported to MySQL Enterprise Service Manager:

- You must ensure that each user configured within your MySQL client application that connects through the agent/proxy and is required to report query analyzer information is allowed to connect to the server from the host on which the agent/proxy is running. When the user connects to the agent/proxy, and the agent/proxy connects to the server the host of the agent/proxy will be used as the identifying client host name during the connection.

  > **Warning**
  >
  > Because the proxy is providing the immediate connectivity to the MySQL server, authentication must use the proxy host-

name, not the client hostname.

To update your user credentials, you need to use the GRANT statement. For example:

```
mysql> GRANT SELECT,UPDATE,INSERT on database.* to 'user'@'localhost' IDENTIFIED BY 'password';
```

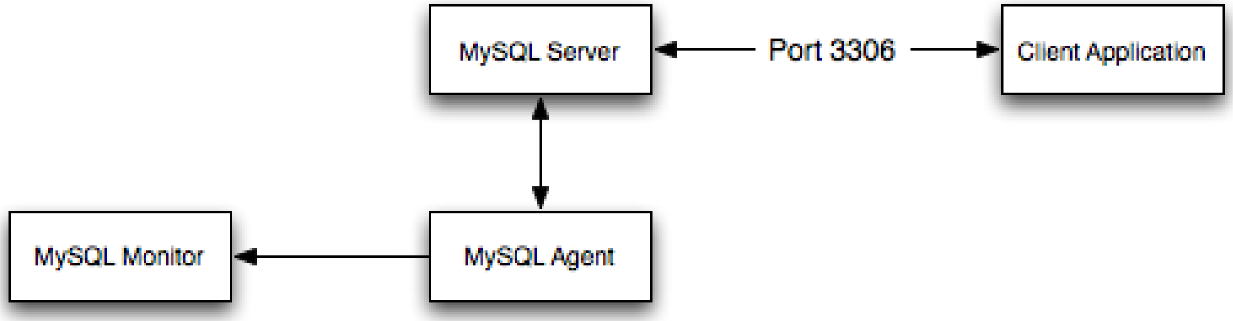
- The MySQL client application user must have SELECT privileges on the mysql.inventory table. This table contains the server UUID and is required to report the query analyzer data to the MySQL Enterprise Service Manager. To enable this, use the GRANT option:

```
mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
```

8.1.1. Enabling Query Analyzer by Changing the MySQL Client Application

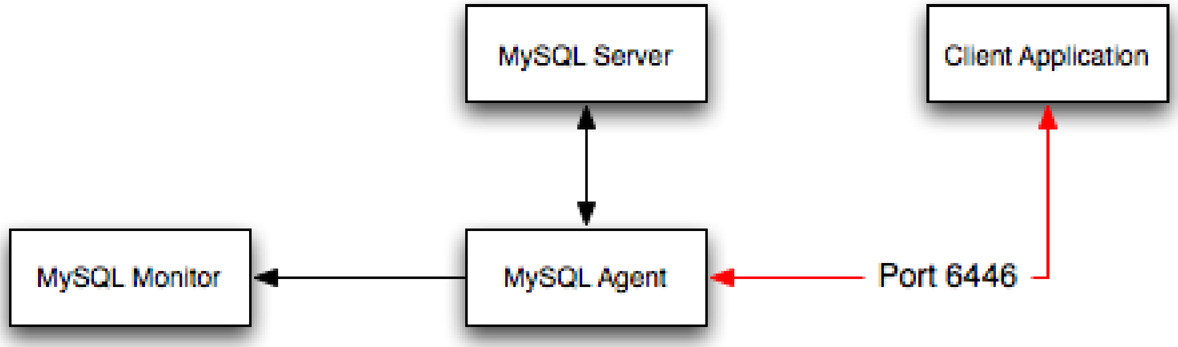
Generally, changing your MySQL client application is the easiest and recommended method. For example, given a typical structure like the one shown in the figure below, the client application would need to be modified so that it no longer communicated directly with the MySQL server, but to the agent/proxy.

Figure 8.2. MySQL Enterprise Dashboard: Standard Agent/Monitor Topology



You can see an example of the structure when communicating using the agent/proxy below.

Figure 8.3. MySQL Enterprise Dashboard: Query Analyzer Agent/Monitor Topology



To enable query analyzer within your MySQL client application:

1. Make sure that the MySQL Enterprise Service Manager and your MySQL Enterprise Monitor Agent are configured and running.

2. Confirm the configuration of your agent by examining the contents of the `etc/mysql-monitor-agent.ini` file within your installed Agent directory.

   Queries will be sent to the host specified in the `proxy-backend-addresses` parameter, and the agent will listen for connections to be redirected to the server on the host name and port configured in the `proxy-address` parameter.

3. Now modify your MySQL client application to communicate with the address specified in the `proxy-address` parameter.

   Alternatively, if you do not want to modify your application directly, you can use iptables or firewall rules to redirect queries from the original host/port combination to the agent's port.

   Because connections to the MySQL server will be coming from the agent/proxy, not the original host, the user credentials used must be have a suitable `GRANT` statement for connections from `localhost`, or the host on which the agent/proxy is executing. The user name and password information will be passed on directly through the agent/proxy from the client to the server.

4. Confirm that your MySQL client application still operates normally. There should be no difference between communicating directly with the MySQL server and communicating using the agent/proxy.

> **Note**
>
> If you are using the `mysql` client to connect to the agent/proxy and your backend servers, make sure that you are communicating with the proxy over the right port. By default, if you specify `localhost` as the host name, then `mysql` will connect using the local Unix domain socket, rather than the TCP/IP socket.
>
> You can enforce `mysql` to use the right port either by explicitly requesting the protocol type, or by using the IP address rather than `localhost`. For example, both of these command lines will start the client using the right protocol:
>
> ```
> shell> mysql --port=4040 --protocol=tcp
> shell> mysql --port=4040 --host=127.0.0.1
> ```

> **Note**
>
> It is recommended that you use one agent/proxy per MySQL server instance. The agent/proxy is not able to forward queries to multiple MySQL server backends.

8.1.2. Enabling Query Analyzer by Changing MySQL Server

When enabling Query Analyzer by changing the MySQL Server, you need to shutdown your server, edit the MySQL configuration file, and then restart MySQL. You will also need to change your Agent/proxy configuration so that the Agent/proxy is listening on the original MySQL TCP/IP port. To use this method:

1. Edit the `/etc/my.cnf` or other MySQL configuration file and change or add the `port` setting from it's current value (default 3306), to another value. For example:

   ```
   port                                            = 3307
   ```

2. Shutdown your MySQL Server.

3. Startup your MySQL Server and confirm that is running.

4. Edit your MySQL Enterprise Monitor Agent configuration so that the agent/proxy is listening for connections on the original MySQL port:

   ```
   proxy-address=:3306
   proxy-backend-addresses = 127.0.0.1:3307
   ```

5. Stop and restart MySQL Enterprise Monitor Agent.

You should now be able to connect to your MySQL server through the MySQL Enterprise Monitor Agent by connecting on the original port:

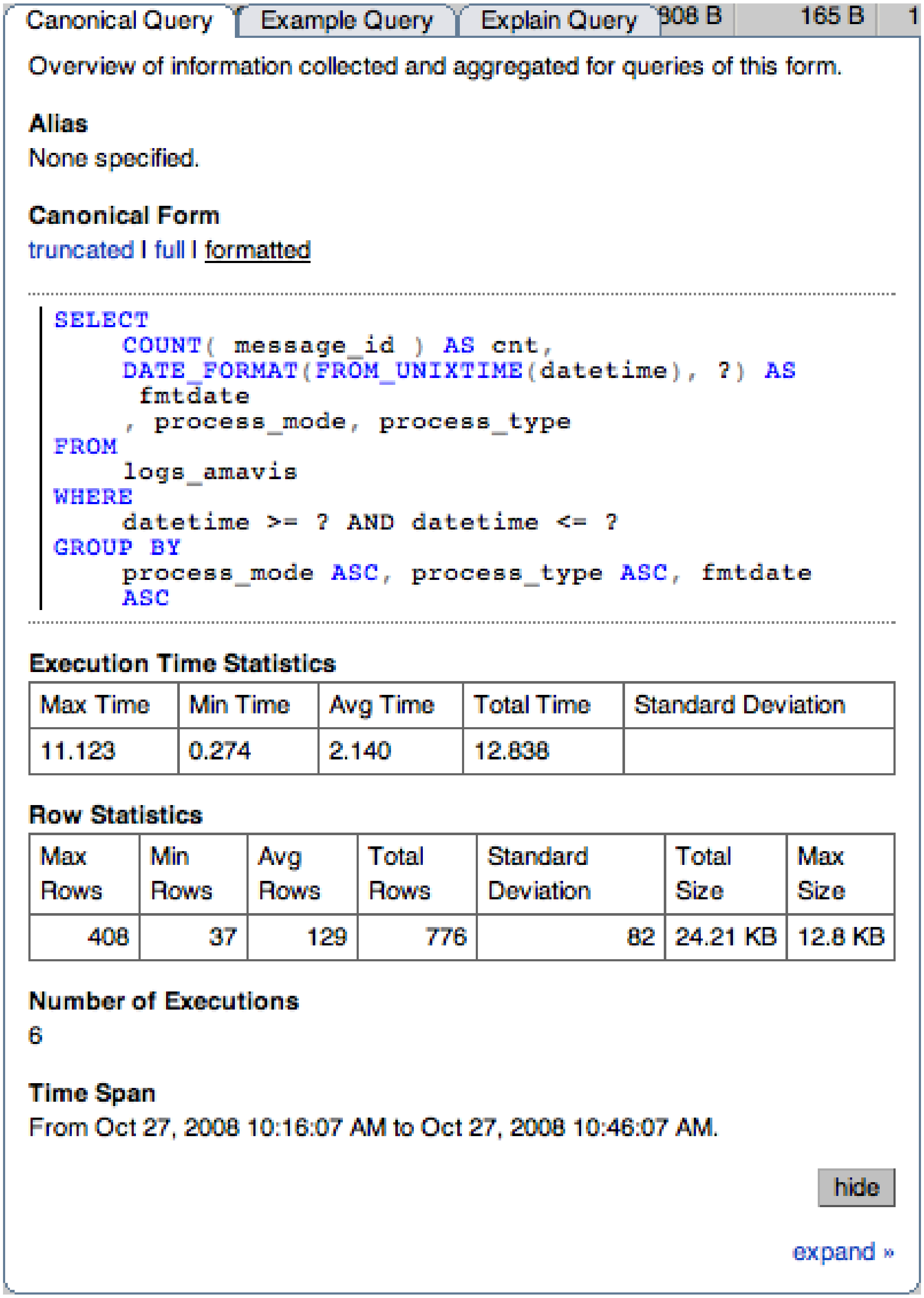
```
shell> mysql --host=127.0.0.1
```

8.2. Getting Detailed Query Information

If you click an individual query, a pop-up window will provide more detailed information about the individual query. You can see an example of this in the figure below. The available tabs within this window will depend on whether you have configured the more detailed query information. By default, you will always be provided the Summary Details page. If enabled, you may also view Example Details, which provide more detailed data about a specific query, including the data and parameters submitted. In addition, you may also enable Example Explain, which provides you with the ability to remotely execute an `EXPLAIN` statement with the specified query and view the resulting information.

- The **Canonical Query** tab:

  Figure 8.4. MySQL Enterprise Dashboard: Canonical Query Tab for a Query



Overview of information collected and aggregated for queries of this form.

Alias

None specified.

Canonical Form

truncated | full | formatted

```
SELECT
     COUNT( message_id ) AS cnt,
     DATE_FORMAT(FROM_UNIXTIME(datetime), ?) AS
      fmtdate
     , process_mode, process_type
FROM
     logs_amavis
WHERE
     datetime >= ? AND datetime <= ?
GROUP BY
     process_mode ASC, process_type ASC, fmtdate
     ASC
```

Execution Time Statistics

Max Time	Min Time	Avg Time	Total Time	Standard Deviation
11.123	0.274	2.140	12.838	

Row Statistics

Max Rows	Min Rows	Avg Rows	Total Rows	Standard Deviation	Total Size	Max Size
408	37	129	776	82	24.21 KB	12.8 KB

Number of Executions

6

Time Span

From Oct 27, 2008 10:16:07 AM to Oct 27, 2008 10:46:07 AM.

In addition to the summary information given in the table, you will get detailed execution statistics, including the minimum time, maximum time, average time, total time and the standard deviation. The standard deviation will enable you to determine whether a particular invocation of a query is outside the normal distribution of times for the given query.

Row statistics provide more detailed contents on the maximum, minimum, average, total, and standard deviation for the number of rows returned by the query, and the total size and maximum size of the data returned. The time period for the total and average figures is shown under the Summary Time Span.

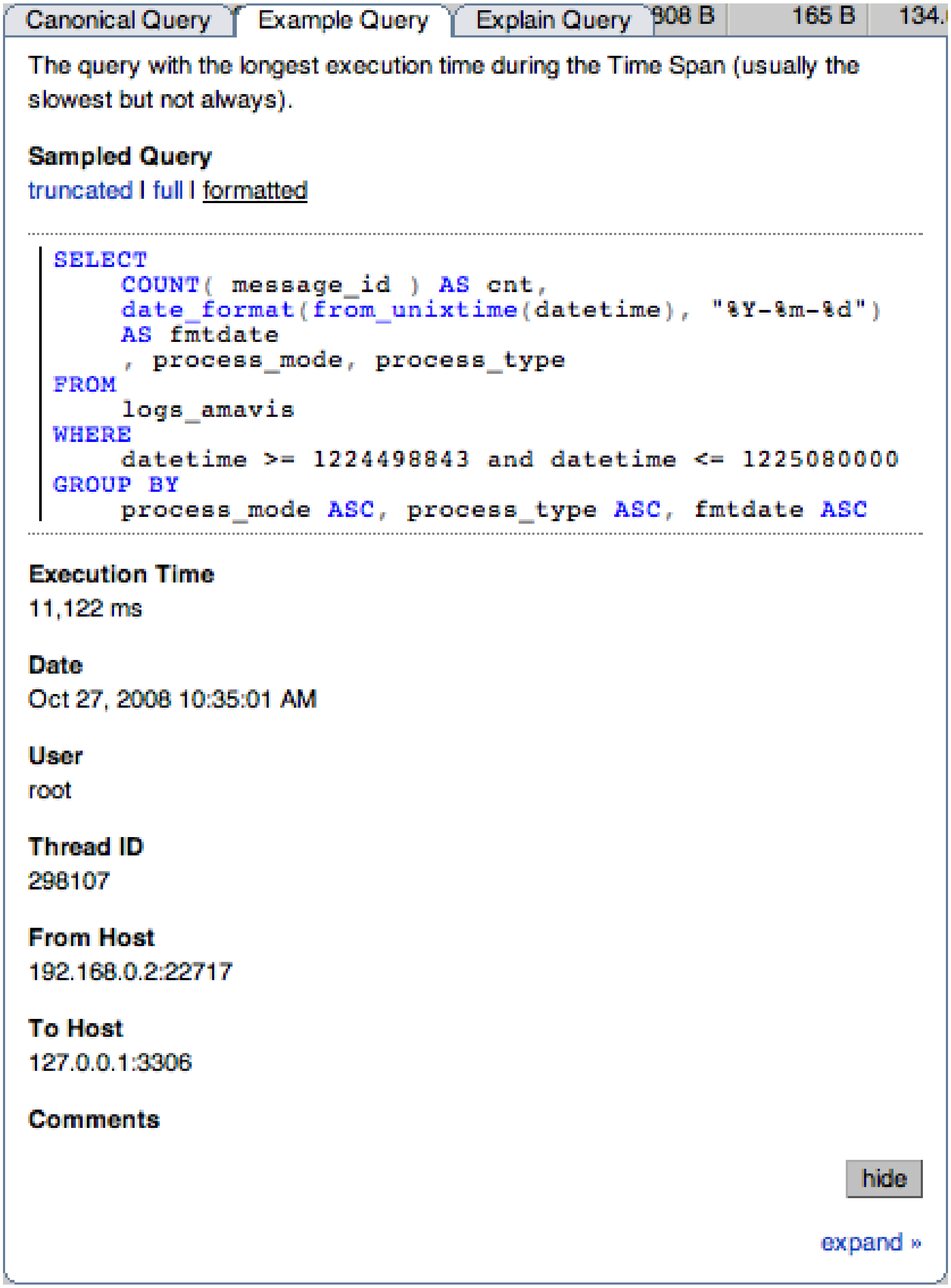
The detailed view for a query also provides three different views of the query. The `truncated` version is a shortened version of the query. The `full` version of the query is the entire query statement. Normalization removes the constants from the individual queries so that queries following the same logical structure are identified as the same basic query.

To close the query detail window, click the HIDE button.

To simplify the identification of a given query, you can create a query alias. The alias will be used in place of the normalized query text within the Query Analyzer table. To create an alias for a query, click the CREATE ALIAS link against the query. The maximum length for a query alias is 255 characters.

- The **Example Query** tab:

Figure 8.5. MySQL Enterprise Dashboard: Example Query Tab for a Query



The query with the longest execution time during the Time Span (usually the slowest but not always).

Sampled Query

truncated | full | formatted

```
SELECT
     COUNT( message_id ) AS cnt,
     date_format(from_unixtime(datetime), "%Y-%m-%d")
     AS fmtdate
     , process_mode, process_type
FROM
     logs_amavis
WHERE
     datetime >= 1224498843 and datetime <= 1225080000
GROUP BY
     process_mode ASC, process_type ASC, fmtdate ASC
```

Execution Time

11,122 ms

Date

Oct 27, 2008 10:35:01 AM

User

root

Thread ID

298107

From Host

192.168.0.2:22717

To Host

127.0.0.1:3306

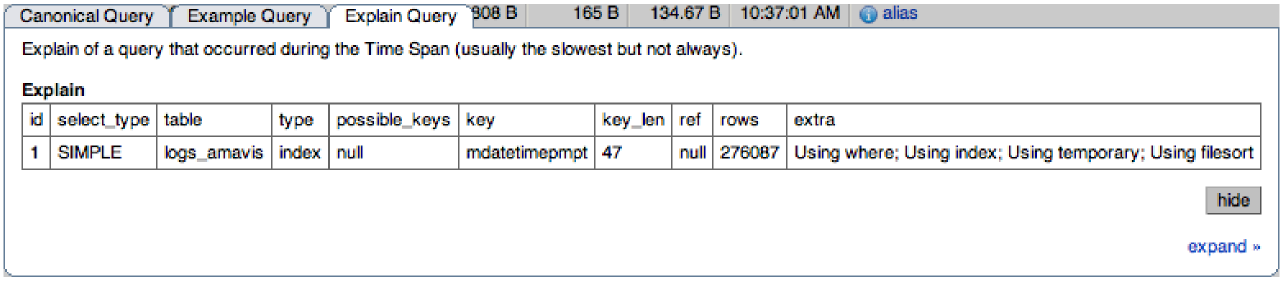
Comments

The Example Details tab provides detailed information about the most expensive query executed, as determined by the execution time.

In addition to the full query, with data, that was executed, the tab shows the execution time, data, user, thread ID, client host and execution host for the given query.

- The **Explain Query** tab:

  Figure 8.6. MySQL Enterprise Dashboard: Explain Query Tab for a Query



Explain of a query that occurred during the Time Span (usually the slowest but not always).

Explain

id	select_type	table	type	possible_keys	key	key_len	ref	rows	extra
1	SIMPLE	logs_amavis	index	null	mdatetimepmpt	47	null	276087	Using where; Using index; Using temporary; Using filesort

  The Example Explain tab enables you to view the output from running the query with the `EXPLAIN` prefix. For more information, see `EXPLAIN` Syntax.

8.3. Filtering Query Analyzer Data

You can filter the queries shown within the Query Analyzer table by using the form at the top of the table. The different fields of the form are used to specify the parameters for the filter process. Once you have specified a filter, all the queries and related statistics shown within the Query Analyzer table are displayed in relation to the filter settings. For example, by default, the filter settings show the queries for the last 30 minutes. All the statistics shown are relative to the last 30 minutes, including average, maximum and execution counts.

The filter fields are:

- **Search Type** and **Query Search** support text searching of the normalized query. For the search type you can specify either a basic text match, or a regular expression match. In addition to the basic text match, you can also search for a query that does not contain a particular string. For regular expression searches, you can specify whether the regular expression should match, or not match (negative regexp) the queries. Regular expressions are parsed using the standard MySQL `REGEXP()` function. For more information, see Regular Expressions.

  > **Note**
  >
  > The search is performed against the canonical version of the query. You cannot search against specific text or values within the parameters of the query itself.

- **Database**: Limit the queries to those executed within a specific database. The database match is performed using the `LIKE` match from the MySQL database, hence you can use the `%` and `_` characters to multiple and single character matches. For more information, see Pattern Matching.

- The **Time Display** menu selects whether the time selection for filtering should be based on the time **interval** (only queries recorded within the displayed time period are shown, using the **Hours** and **Minutes** pop-up), or whether the selection should be based on a time period (**From/To**), where you can select the time range to be displayed.

  Using the **Interval** mode shows queries within the given time period from the point the graph was updated. For example, if you select 30 minutes, then the queries shown were captured within the last 30 minutes. If you updated the display at 14:00, then the queries displayed would have been captured between 13:30 and 14:00. Using interval mode limits the timespan for the filter selection to a maximum of 23 hours and 59 minutes.

  Using the **From/To** mode enables you to show queries between specific dates and times. Using this mode you can show only the queries received during a specific time span, and you can display the query history for a much longer time period, for as long as you have been recording query analysis information.

- The **View** selection determines whether the information should be returned on a group basis, where an aggregate of the same query executed on all monitored servers is shown, or on a **Server** basis, where queries are summarized by individual server. If the latter option has been selected, the table includes an additional column showing the server.

- **Query Type** lets you select the type of query on which to filter queries. Selecting **All** will show all queries. Additional query types you can select include `SELECT`, `INSERT`, `UPDATE` and other main SQL query types.

- **LIMIT** specifies the number of queries to be displayed within each page.

When you have set your filter parameters, you can update the Query Analysis display by clicking the FILTER button. To reset the fields to the default settings click the RESET button.

8.4. Using Query Analyzer Data

The information provided by Query Analyzer can be complex to understand and resolve into simple targets and resolutions for your MySQL client application. The information can be used in different ways to find problems in your queries or your servers, or both. Provided below are some tips on how to get the best out of the Query Analyzer interface, and how to identify different queries and problems based on the information shown by the Query Analyzer system.

First, consider the information provided by individual columns by your queries. In particular, the following columns can highlight specific problems with your queries or database server:

- **Execution Count**: High execution counts, especially for a query that you expect to be executed very rarely, may indicate that your MySQL client application is either running a simple query to frequently, or may be running a query multiple times that could otherwise be cached. You should pay particular attention to queries where the execution count increases significantly in a short period of time compared to the normal execution rate.

  How to find: Use the sort feature to sort the queries by execution count.

- **New queries**: New queries appearing in the Query Analyzer tab, especially if they appear after other queries have been in the display for a number of hours or days may indicate a number of issues:

- **Execution times**: Long execution times, and a long max execution time compared to the average execution time may indicate a problem with a specific query and specific parameters.

  How to find: Use the sort feature to sort the queries by execution count.

You can also use the filtering and sort options to get specific information about potential problem queries.

8.5. Troubleshooting Query Analyzer

If you are having trouble with Query Analyzer, either because the information is not being shown or the full range of queries that you expect are not appearing in the Query Analyzer page then there are a number of systems you can check.

To confirm that your system is correctly configured for Query Analysis, check the following:

- Confirm that the agent is running by checking the Agent log and the status of the server within MySQL Enterprise Service Manager

- Check the configuration of the agent. You must confirm the following:

  - The `plugins` parameter within the main configuration file, `mysql-monitor-agent.ini`, must contain the `proxy` plugin:

    ```
    plugins=proxy,agent
    ```

  - The `agent-item-files` parameter within the main configurationfile, `mysql-monitor-agent.ini`, must specify the `share/mysql-proxy/items/quan.lua` script:

    ```
    agent-item-files = share/mysql-proxy/items/quan.lua, »
         share/mysql-proxy/items/items-mysql-monitor.xml
    ```

  - The proxy configuration parameters must point to the MySQL server where you want your queries to be sent. For example, if you are running your agent on the same host as your MySQL server then you might have the following lines in your `mysql-monitor-agent.ini` file:

```
proxy-address=:4040
proxy-backend-addresses = 127.0.0.1:3306
proxy-lua-script        = share/mysql-proxy/quan.lua
```

The above configuration can means:

- The agent/proxy will listen on the current machine, using port 4040 (`proxy-address`).
- The agent/proxy will send all queries received on to the host `127.0.0.1` on port `3306` (the standard MySQL port), as per the `proxy-backend-addresses` parameter.

You can see a sample complete configuration file (`mysql-monitor-agent.ini`), using the `127.0.0.1` as the MySQL backend server, and reporting to a MySQL Enterprise Service Manager called `monitor`:

```
[mysql-proxy]

plugins=proxy,agent
agent-mgmt-hostname = http://agent:password@monitor:18080/heartbeat
mysqld-instance-dir= etc/instances
agent-item-files = share/mysql-proxy/items/quan.lua,share/mysql-proxy/items/items-mysql-monitor.xml
proxy-address=:4040
proxy-backend-addresses = 127.0.0.1:3306
proxy-lua-script        = share/mysql-proxy/quan.lua


agent-uuid = a3113263-4993-4890-8235-cadef9617c4b
log-file = mysql-monitor-agent.log
pid-file=/opt/mysql/enterprise/agent/mysql-monitor-agent.pid
```

- Confirm that you can connect through the agent proxy to your backend MySQL server. You can do this by checking with the MySQL client. You must specify the same options as you would if you were connecting to the original server, including specifying the same user and password information:

```
shell> mysql -h 127.0.0.1 --port 4040 --user=root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 299239
Server version: 5.0.60-log Gentoo Linux mysql-5.0.60-r1

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

mysql>
```

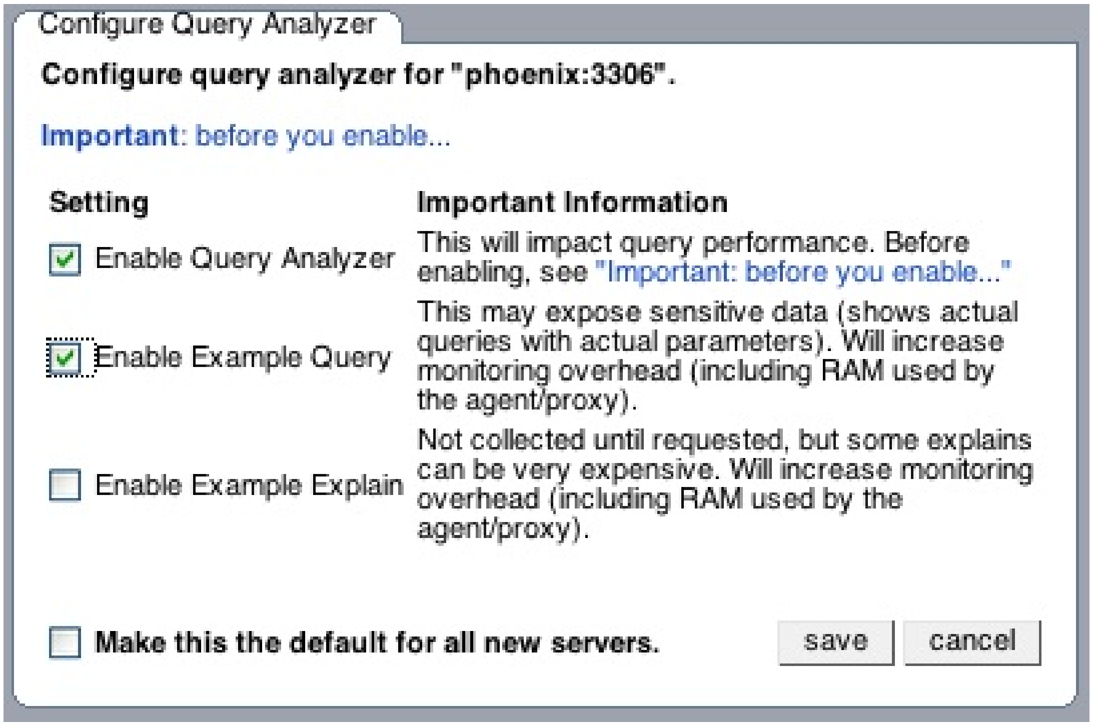
```
shell> mysql -h 127.0.0.1 --port 6446 --user=root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 299239
Server version: 5.0.60-log Gentoo Linux mysql-5.0.60-r1

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

mysql>
```

- Check that your MySQL client application is configured to use the configured proxy port, instead of the real MySQL port when sending queries.
- Confirm that Query Analyzer enabled for your host. For more information, see Section 8.6, “Query Analyzer Settings”.

8.6. Query Analyzer Settings

There are a number of settings related to the Query Analyzer data. You can configure the query analyzer operation by using the **CONFIGURE QUERY ANALYZER** link within the **QUERY ANALYZER** tab, or through the CONFIGURE QUERY ANALYZER button within the **MANAGE SERVERS** tab within the **SETTINGS** tab. Both methods provide you with the same dialog:

Figure 8.7. MySQL Enterprise Dashboard: Query Analyzer Configuration



Configure Query Analyzer

Configure query analyzer for "phoenix:3306".

Important: before you enable...

Setting	Important Information
☑ Enable Query Analyzer	This will impact query performance. Before enabling, see "Important: before you enable..."
☑ Enable Example Query	This may expose sensitive data (shows actual queries with actual parameters). Will increase monitoring overhead (including RAM used by the agent/proxy).
□ Enable Example Explain	Not collected until requested, but some explains can be very expensive. Will increase monitoring overhead (including RAM used by the agent/proxy).

□ **Make this the default for all new servers.**

Through either solution, the configuration options that you select are applied to the individual server or server group selected within the **SERVERS**navigation panel.

There are three configuration options available through either method:

- **ENABLE QUERY ANALYZER** configures whether query analyzer should be enabled for this server or server group. If selected, query analyzer will be enabled. To disable, delect the check box.

  If Query Analyzer has been enabled, then you can additional configure the Example Query function by selecting the **ENABLE EXAMPLE QUERY** checkbox. Enabling this option provides an additional tab when you open the **CANONICAL QUERY** window when clicking a query.

- **ENABLE EXAMPLE QUERY** allows the Query Analyzer to display more information about individual queries. When enabled, queries and their data items (rather than the canonical form shown by default) will be provided. Enabling this option may expose the full query statements and therefore may present a security issue.

  With the **EXAMPLE QUERY** option enabled, an additional tab within the query summary details is made available. For more information, see Section 8.2, “Getting Detailed Query Information”.

  If you have enabled **EXAMPLE QUERY**, then you can additional enable **EXAMPLE EXPLAIN**, To enable this tab, select the **ENABLE EXAMPLE EXPLAIN** checkbox.

- **ENABLE EXAMPLE EXPLAIN** provides another tab when viewing a query where you can view the output from `EXPLAIN` output from MySQL for the selected query. This will show the full query and how the query was executed within the servers.

  Enabling this option may add overhead to the execution of your server, as the server will run an `EXPLAIN` statement each time it identifies a long running query. For more information, Appendix B, *MySQL Enterprise Monitor Frequently Asked Questions*.

To enable or disable query analyzer for an individual server, go to the **SETTINGS** page and click the MANAGE SERVERS link. To configure all the properties, click the **CONFIGURE QUERY ANALYZER** link next to server you want modify.

Alternatively, for each server, the **QUERY ANALYZER** column shows the current setting, On or Off, and whether the **EXAMPLE** and **EXPLAIN** functionality is enabled. To change any setting, click the current status to toggle between the On/Off position.

To disable or enable Query Analyzer for the selected servers, use the DISABLE QUERY ANALYZER or `enable query analyzer` buttons within the **SETTINGS** page. You must have selected one or more servers from the list of available servers before selecting these buttons.

You can use the options that you have just selected as the default for all new servers that register with MySQL Enterprise Service Manager by using select the **MAKE THIS THE DEFAULT FOR ALL NEW SERVERS** checkbox. By default, when a new server registers with MySQL Monitor, the server is automatically configured to supply Query Analyzer data. This can have impact on the performance of your monitor and agent as it increases the amount of information supplied to the MySQL Monitor.

Configuration of Query Analyzer occurs through the CONFIGURE DEFAULTS button from within the **QUERY ANALYZER** page.

Chapter 9. The Replication Page

Note

MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

Navigate to the `Replication` page by choosing the **Replication** tab. This page provides a quick summary view of the state of your replication servers or, if you wish, you can drill down and determine specifics about any master or slave.

Note

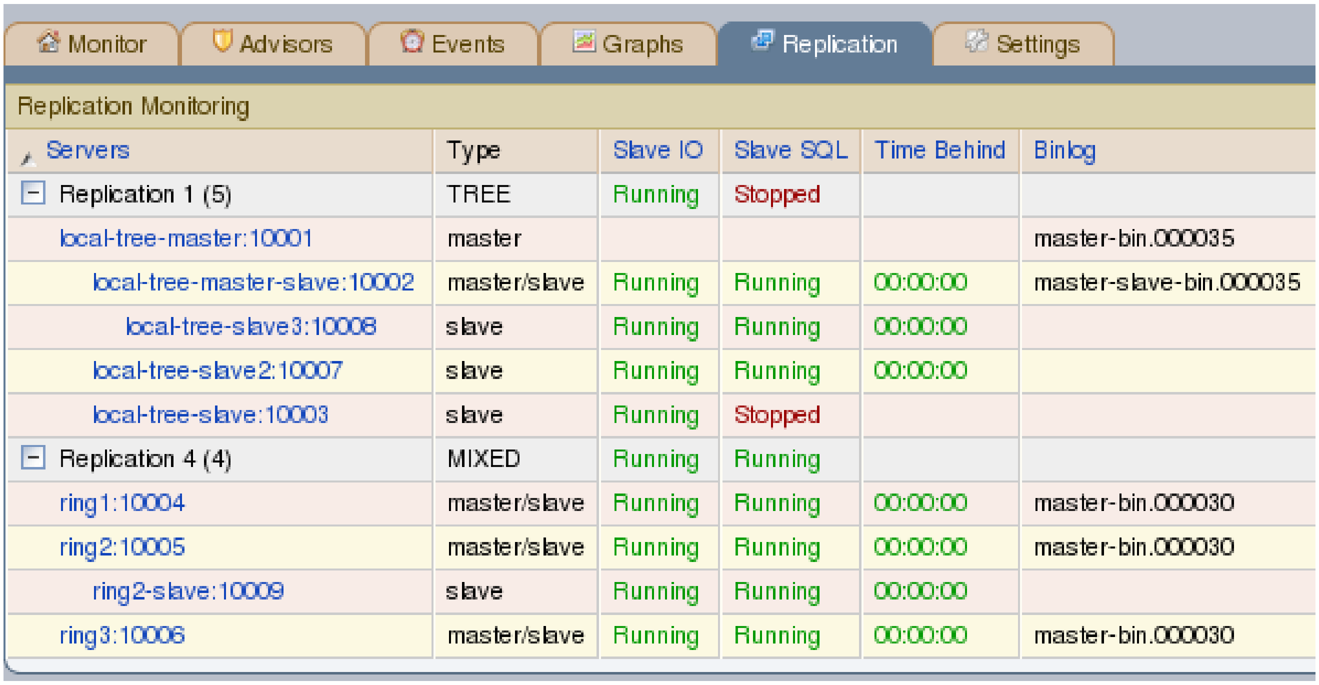
Servers, whether masters or slaves, must be monitored for them to appear on this page.

Note

There will be no `Replication` page if your subscription level does not support this feature.

The `Replication` page groups all master servers with their slaves. Masters and their slaves are autodiscovered and a grouping is created. This grouping shows up on the replication page and also in the `Heat Chart` on the `Monitor` page. Scans run on a five minute interval, so depending upon the order of discovery, it can take as long as 2 polling intervals to create a complete group.

Figure 9.1. MySQL Enterprise Dashboard: Replication Groups



Servers	Type	Slave IO	Slave SQL	Time Behind	Binlog
⊟ Replication 1 (5)	TREE	Running	Stopped		
local-tree-master:10001	master				master-bin.000035
local-tree-master-slave:10002	master/slave	Running	Running	00:00:00	master-slave-bin.000035
local-tree-slave3:10008	slave	Running	Running	00:00:00	
local-tree-slave2:10007	slave	Running	Running	00:00:00	
local-tree-slave:10003	slave	Running	Stopped		
⊟ Replication 4 (4)	MIXED	Running	Running		
ring1:10004	master/slave	Running	Running	00:00:00	master-bin.000030
ring2:10005	master/slave	Running	Running	00:00:00	master-bin.000030
ring2-slave:10009	slave	Running	Running	00:00:00	
ring3:10006	master/slave	Running	Running	00:00:00	master-bin.000030

Discovery events are logged to the `Replication` log. To view this log navigate to the `Settings` page and choose the **Logs** link. View all replication-related events by clicking the **Replication** link. This log can be a useful tool should you need to debug the replication topology discovery process.

Warning

The agent must be installed on the same machine as the server you are monitoring for discovery to work properly. Do **not** use remote monitoring.

Replication groups can be managed from the `Manage Servers` page in the same way as other groups. However, any slaves removed from a server group will automatically be restored to that group. It is also possible to add nonslaves to a replication grouping. For more information about server groupings see Section 4.3.2, “Grouping Servers”.

9.1. Replication Page Details

Choose a value from the **REFRESH** drop-down list box to set the rate at which information is updated. This refresh rate applies only to the information presented on this page: It is independent of the rate set for the `Monitor` page.

The following columns describe replication servers and their slaves:

- Servers: Displays the group name and any master servers and slaves
- Type: Indicates the topology of a server group or in the case of individual servers, whether a server is a master, a master/slave, or a slave
- Slave IO: Reports the status of the slave I/O thread
- Slave SQL: Reports the status of the slave SQL thread
- Seconds Behind: The number of seconds the slave is behind the master. This column is blank if a server is a master.
- Binlog: The binary log file name
- Binlog Pos: The current position in the binary log file
- Master Binlog: The master binary log file name
- Master Binlog Pos: The current position in the master binary log file
- Last Error: The most recent error
- Unlabeled Column: Use the **RENAME GROUP** link on the server group line to edit the server group name

Levels of indentation in the `Servers` column show the relationship between master servers and their slaves. Most column headings are active links that allow you to change the order of display by clicking the header. Sorting works differently for different column groupings. Click the `Seconds Behind` header to order servers by the number of seconds they are behind their master. However, in all cases, the server topology is respected. For example, in a `TREE` topology, ordering occurs within branches only.

If the agent is down, servers show in bold red in the `Servers` column. The `Slave IO` and the `Slave SQL` columns display `stopped` in red text if these threads are not running. If an agent is down, italics is used to display the last know status of the I/O or SQL threads.

Clicking a master server opens a dialog box that displays information about the server. The information shown includes:

- The number of slave servers
- The binary log file name
- The binary log position
- Which databases are replicated and which not

The dialog box also includes a link that allows the user to hide or show the slave servers.

Clicking a slave server opens a dialog window showing extensive information about the slave.

Appendix A. Licenses for Third-Party Components

MySQL Enterprise Monitor 2.0

A.1. Apache Commons Collections License

The following software may be included in this product:

```
Apache Commons Collections

Component's NOTICE.txt file:
Apache Commons Collections
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.2. Apache Commons DBCP License

The following software may be included in this product:

```
Apache Commons DBCP

Component's NOTICE.txt file:
Apache Commons DBCP
Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.3. Apache Commons Digester License

The following software may be included in this product:

```
Apache Commons Digester

Component's NOTICE.txt file:
Apache Jakarta Commons Digester
Copyright 2001-2006 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.4. Apache Commons FileUpload License

The following software may be included in this product:

```
Apache Commons FileUpload

Component's NOTICE.txt file:
Apache Jakarta Commons FileUpload
Copyright 2002-2006 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.5. Apache Commons IO License

The following software may be included in this product:

```
Apache Commons IO

Component's NOTICE.txt file:
Apache Jakarta Commons IO
Copyright 2001-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.6. Apache Commons Lang License

The following software may be included in this product:

```
Apache Commons Lang

Component's NOTICE.txt file (older version):
Apache Jakarta Commons Lang
Copyright 2001-2007 The Apache Software Foundation

This product includes software developed by
```

```
The Apache Software Foundation (http://www.apache.org/).

Component's NOTICE.txt file (newer version):
Apache Commons Lang
Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.7. Apache Commons Logging License

The following software may be included in this product:

```
Apache Commons Logging

Component's NOTICE.txt file:
Apache Commons Logging
Copyright 2003-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.8. Apache Commons Pool License

The following software may be included in this product:

```
Apache Commons Pool

Component's NOTICE.txt file:
Apache Commons Pool
Copyright 1999-2009 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.9. Apache Commons Validator License

The following software may be included in this product:

```
Apache Commons Validator

Component's NOTICE.txt file:
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.10. Apache License Version 2.0, January 2004

```
The following applies to all products licensed under the Apache 2.0
License: You may not use the identified files except in compliance
with the Apache License, Version 2.0 (the "License.") You may obtain a
copy of the License at http://www.apache.org/licenses/LICENSE-2.0. A
copy of the license is also reproduced below. Unless required by
applicable law or agreed to in writing, software distributed under the
License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied. See the License for
the specific language governing permissions and limitations under the
License.

Apache License Version 2.0, January 2004 http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
```

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

```
(c) You must retain, in the Source form of any Derivative Works that
You distribute, all copyright, patent, trademark, and attribution
notices from the Source form of the Work, excluding those notices that
do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain
to any part of the Derivative Works, in at least one of the following
places: within a NOTICE text file distributed as part of the
Derivative Works; within the Source form or documentation, if provided
along with the Derivative Works; or, within a display generated by the
Derivative Works, if and wherever such third-party notices normally
appear. The contents of the NOTICE file are for informational purposes
only and do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside or as
an addendum to the NOTICE text from the Work, provided that such
additional attribution notices cannot be construed as modifying the
License.

You may add Your own copyright statement to Your modifications and may
provide additional or different license terms and conditions for use,
reproduction, or distribution of Your modifications, or for any such
Derivative Works as a whole, provided Your use, reproduction, and
distribution of the Work otherwise complies with the conditions stated
in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work by
You to the Licensor shall be under the terms and conditions of this
License, without any additional terms or conditions.  Notwithstanding
the above, nothing herein shall supersede or modify the terms of any
separate license agreement you may have executed with Licensor
regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed
to in writing, Licensor provides the Work (and each Contributor
provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied, including, without
limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT,
MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely
responsible for determining the appropriateness of using or
redistributing the Work and assume any risks associated with Your
exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise, unless
required by applicable law (such as deliberate and grossly negligent
acts) or agreed to in writing, shall any Contributor be liable to You
for damages, including any direct, indirect, special, incidental, or
consequential damages of any character arising as a result of this
License or out of the use or inability to use the Work (including but
not limited to damages for loss of goodwill, work stoppage, computer
failure or malfunction, or any and all other commercial damages or
losses), even if such Contributor has been advised of the possibility
of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer, and
charge a fee for, acceptance of support, warranty, indemnity, or other
liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only on
Your own behalf and on Your sole responsibility, not on behalf of any
other Contributor, and only if You agree to indemnify, defend, and
hold each Contributor harmless for any liability incurred by, or
claims asserted against, such Contributor by reason of your accepting
any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate
notice, with the fields enclosed by brackets "[]" replaced with your own
identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also
recommend that a file or class name and description of purpose be included
on the same "printed page" as the copyright notice for easier identification
within third-party archives.

Copyright [yyyy] [name of copyright owner]
```

```
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied. See the License for the specific language governing permissions
and limitations under the License.
```

A.11. Apache log4j License

The following software may be included in this product:

```
Apache log4j

Component's NOTICE.txt file:
Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.12. Apache Tomcat License

The following software may be included in this product:

```
Apache Tomcat

Component's NOTICE.txt file:
Apache Tomcat
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
```

This component is licensed under Section A.10, “Apache License Version 2.0, January 2004”.

A.13. cURL (`libcurl`) License

The following software may be included in this product:

```
cURL (libcurl)

Use of any of this software is governed by the terms of the license below:

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2009, Daniel Stenberg, <daniel@haxx.se>.
All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY
RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT
OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other
dealings in this Software without prior written authorization of the copyright
holder.
```

A.14. JDOM Project License

The following software may be included in this product:

```
JDOM

This product includes software developed by the
JDOM Project (http://www.jdom.org/).

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright notice,
   this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions, and the disclaimer that follows these conditions
   in the documentation and/or other materials provided with the distribution.
3. The name "JDOM" must not be used to endorse or promote products derived
   from this software without prior written permission. For written permission,
   please contact <request_AT_jdom_DOT_org>.
4. Products derived from this software may not be called "JDOM", nor may
   "JDOM" appear in their name, without prior written permission from the
   JDOM Project Management <request_AT_jdom_DOT_org>.
In addition, we request (but do not require) that you include in the end-user
documentation provided with the redistribution and/or in the software itself
an acknowledgement equivalent to the following:
   "This product includes software developed by the
   JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos available
at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

This software consists of voluntary contributions made by many individuals
on behalf of the JDOM Project and was originally created by Jason Hunter
and Brett McLaughlin. For more information on the JDOM Project, please see
<http://www.jdom.org/>."
```

A.15. jQuery License

The following software may be included in this product:

```
jQuery

Copyright (c) 2009 John Resig.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
```

A.16. jQuery UI License

The following software may be included in this product:

```
jQuery UI

Copyright (c) 2009 Paul Bakaus, http://jqueryui.com

This software consists of voluntary contributions made by many individuals
```

```
(AUTHORS.txt, http://jqueryui.com/about) For exact contribution history,
see the revision history and logs, available at
http://jquery-ui.googlecode.com/svn/

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

A.17. `libevent` License

The following software may be included in this product:

```
libevent

Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE
```

A.18. `LPeg` Library License

The following software may be included in this product:

```
LPeg

Use of any of this software is governed by the terms of the license below:

Copyright © 2008 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

A.19. LuaFileSystem Library License

The following software may be included in this product:

```
LuaFileSystem

Copyright © 2003 Kepler Project.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

A.20. OpenSSL v0.9.8i License

The following software may be included in this product:

```
OpenSSL v0.9.8i

Copyright (c) 1998-2006 The OpenSSL Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the
   distribution.
3. All advertising materials mentioning features or use of this
   software must display the following acknowledgment:
      "This product includes software developed by the OpenSSL
       Project for use in the OpenSSL Toolkit.
       (http://www.openssl.org/)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be
   used to endorse or promote products derived from this software
   without prior written permission. For written permission, please
   contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL"
   nor may "OpenSSL" appear in their names without prior written
   permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following
   acknowledgment:
      "This product includes software developed by the OpenSSL
       Project for use in the OpenSSL Toolkit
       (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com). This product includes software written by Tim
Hudson (tjh@cryptsoft.com).
```

A.21. StringTemplate Template Engine License

The following software may be included in this product:

```
StringTemplate Template Engine

Copyright (c) 2008, Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the distribution.
3. Neither the name of the author nor the names of its contributors may
   be used to endorse or promote products derived from this software
   without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
```

A.22. `zlib` License

The following software may be included in this product:

zlib

Oracle gratefully acknowledges the contributions of Jean-loup Gailly and Mark Adler in creating the zlib general purpose compression library which is used in this product.

```
/* zlib.h -- interface of the 'zlib' general purpose compression library version 1.2.3,
July 18th, 2005
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty.
In no event will the authors be held liable for any damages arising from the
use of this software. Permission is granted to anyone to use this software
for any purpose,including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:
1. The origin of this software must not be misrepresented; you must not claim
that you wrote the original software. If you use this software in a product,
an acknowledgment in the product documentation would be appreciated but is not
required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org
Mark Adler madler@alumni.caltech.edu
*/
```

Appendix B. MySQL Enterprise Monitor Frequently Asked Questions

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

FAQ Categories

- **General Usage**
- **MySQL Enterprise**
- **MySQL Monitor**
- **MySQL Query Analyzer**

General Usage

Questions

- B.1: While monitoring my network traffic I have noticed that the agents communicate information at irregular intervals to the MySQL Enterprise Service Manager. I cannot see anything in my configuration that would explain this behavior. What is going on?
- B.2: My MySQL Enterprise Service Manager is behind a firewall but it cannot communicate with the MySQL Enterprise website to register and download my license key and advisor bundle. I normally use a proxy service to access external websites. How do I configure the proxy settings for MySQL Enterprise Dashboard?
- B.3: How do I change the name of a server?
- B.4: I have started a Data Migration of my old data for a server to MySQL Enterprise Service Manager 2.0, but I have noticed that the performance of the monitor server has degraded significantly. Can I stop the migration?
- B.5: I have set the graphs to update every 5 minutes, and the page refresh to occur every minute. The page is refreshing correctly, but the graphs do not seem to update.
- B.6: During query analysis, I am unable to obtain an `EXAMPLE` or `EXPLAIN` information when examining the detail of the analyzed query within the **QUERY ANALYZER** panel.
- B.7: I have enabled `EXPLAIN` queries for Query Analyzer, but no queries with the `EXPLAIN` data are showing up in the display.
- B.8: Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?
- B.9: Why do some rules appear to have a **SEVERITY** of **UNKNOWN**?
- B.10: What is the relationship between the advisor JAR file and the key?
- B.11: Does the Gold-level key support Silver-level advisors?
- B.12: Can the Trial-level key work with the Gold-level advisors JAR file?
- B.13: Can I run MySQL Enterprise Service Manager on machine with other applications running?
- B.14: How frequently is the data purge process executed?

Questions and Answers

B.1: While monitoring my network traffic I have noticed that the agents communicate information at irregular intervals to the MySQL Enterprise Service Manager. I cannot see anything in my configuration that would explain this behavior. What is going

on?

Each MySQL Enterprise Monitor Agent periodically sends information to the server about a range of different information, including the core rule and statistical data, Query Analyzer information and other data used to monitor the status of your MySQL server.

One element of this is called the MySQL Enterprise Monitor Agent Heartbeat, which is core information exchange that indicates that the monitored server is still up and running. The heartbeat information is vital because it tells the MySQL Enterprise Service Manager that the agent and server are still communicating. This information is sent regularly to the MySQL Enterprise Service Manager, but to prevent multiple agents from sending the information at the same time, and creating a large network load (or storm), the interval is randomized with each heartbeat. The randomization ensures that the information is still uploaded periodically, but without the potential to overload the network with this data.

B.2: My MySQL Enterprise Service Manager is behind a firewall but it cannot communicate with the MySQL Enterprise website to register and download my license key and advisor bundle. I normally use a proxy service to access external websites. How do I configure the proxy settings for MySQL Enterprise Dashboard?

To configure a proxy service, you need to edit the `apache-tomcat/conf/catalina.properties` file within the MySQL Enterprise Service Manager installation directory. To make the changes, the proxy configuration information to the end of the file by setting the `http.proxyHost` and `http.proxyPort` properties:

```
http.proxyHost=proxy.example.com
http.proxyPort=8080
```

You will need to restart the MySQL Enterprise Service Manager for the change to take effect:

```
shell> mysqlmonitorctl.sh restart
```

B.3: How do I change the name of a server?

Go to the **Manage Servers** panel within **Settings** and click Rename server.

Renaming the server in this way will override all other server naming, including changes to the agent configuration.

B.4: I have started a Data Migration of my old data for a server to MySQL Enterprise Service Manager 2.0, but I have noticed that the performance of the monitor server has degraded significantly. Can I stop the migration?

You can stop the migration of your historical data at any time. Go to the **Manage Servers** display of the **Settings** panel and click Stop next to each server that is being migrated. You can restart the migration at any point.

B.5: I have set the graphs to update every 5 minutes, and the page refresh to occur every minute. The page is refreshing correctly, but the graphs do not seem to update.

The graph refresh and page refresh are two different parameters. The graphs will update according to their refresh period, regardless of the refresh period set for the main display page.

B.6: During query analysis, I am unable to obtain an `EXAMPLE` or `EXPLAIN` information when examining the detail of the analyzed query within the Query Analyzer panel.

You must explicitly enable the `EXAMPLE` and `EXPLAIN` query functionality. Make sure that you have enabled both panels. See Section 8.6, “Query Analyzer Settings”.

B.7: I have enabled `EXPLAIN` queries for Query Analyzer, but no queries with the `EXPLAIN` data are showing up in the display.

Query Analyzer only obtains `EXPLAIN` information when the MySQL Enterprise Monitor Agent identifies a long running query. If none of your queries exceed the defined threshold, then the `EXPLAIN` information is not obtain and provided to the Query Analyze for display.

To change the query duration at which an `EXPLAIN` is triggered, you must edit the `share/mysql-proxy/quan.lua` file within the MySQL Enterprise Monitor Agent directory on each server. You need to change the value configured in the `auto_explain_min_exec_time_us`. The default is 500ms:

```
---
-- configuration
--
-- SET GLOBAL analyze_query.auto_filter = 0
if not proxy.global.config.quan then
        proxy.global.config.quan = {
                analyze_queries  = true,   -- track all queries
                query_cutoff     = 160,    -- only show the first 160 chars of the query
                num_worst_queries = 5,
                auto_explain     = true,
                auto_explain_min_exec_time_us = 500 * 1000
        }
end
```

The value is expressed in microseconds, which is why the value must be multiplied by 1000. To reduce this value to 100ms you would modify the line:

```
auto_explain_min_exec_time_us = 100 * 1000
```

You do not need to restart MySQL Enterprise Monitor Agent for the changes to take effect.

B.8: Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?

The MySQL Proxy component, and Query Analyzer, require that clients connecting through MySQL Enterprise Monitor Agent are using MySQL 5.0 or later. Clients that use the library provided with MySQL 4.1 or earlier will not work with MySQL Enterprise Monitor Agent.

B.9: Why do some rules appear to have a SEVERITY of UNKNOWN?

Due to timing issues, certain rules such as “32-Bit Binary Running on 64-Bit AMD Or Intel System” and “Key Buffer Size Greater Than 4 GB” do not evaluate correctly due to timing issues. This is a known issue and will be resolved in future versions of MySQL Enterprise Monitor.

B.10: What is the relationship between the advisor JAR file and the key?

The JAR file contains graph and advisor definitions. The key file contains typical customer validation data such as contract information, number of servers covered, subscription level and dates.

B.11: Does the Gold-level key support Silver-level advisors?

The Gold-level advisor JAR file will contain Silver-level advisors plus Gold-level advisors. However, you cannot use the Gold-level key with the Silver-level advisors JAR file. The Gold-level key can only be used with the Gold-level advisors JAR file.

B.12: Can the Trial-level key work with the Gold-level advisors JAR file?

The Trial-level key can only be used with the Trial-level advisors JAR file.

B.13: Can I run MySQL Enterprise Service Manager on machine with other applications running?

You can, but ideally you should be running your MySQL Enterprise Service Manager on a dedicated machine, especially if you are monitoring a number of different agents. For more information, see Section F.5, “Choosing Suitable MySQL Enterprise Service Manager Hardware Configurations”.

B.14: How frequently is the data purge process executed?

A data purge process is started approximately once a minute. If you have changed the purge period then the data will start to be purged within the next minute.

MySQL Enterprise

Questions

- B.1: What is MySQL Enterprise?
- B.2: Can I buy MySQL Enterprise subscriptions for multiple years?
- B.3: Can I buy MySQL Enterprise subscriptions for only some of my production MySQL database servers?
- B.4: Do all my MySQL Enterprise subscriptions need to be at the same tier?
- B.5: What if I plan to add more MySQL servers to my MySQL Enterprise subscription?
- B.6: How should I decide between MySQL Enterprise Basic, Silver, Gold and Platinum?

Questions and Answers

B.1: What is MySQL Enterprise?

The MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services and production support to ensure your business achieves the highest levels of reliability, security and uptime.

MySQL Enterprise includes:

- MySQL Enterprise Server software, the most reliable, secure and up-to date version of the world’s most popular open source database
- MySQL Enterprise Monitor that continuously monitors your database and proactively advises you on how to implement MySQL best practices
- MySQL 24x7 Production Support with fast response times to assist you in the development, deployment and management of MySQL applications

MySQL Enterprise is available in 4 tiers (Basic, Silver, Gold, Platinum). http://www.mysql.com/products/enterprise/features.html

B.2: Can I buy MySQL Enterprise subscriptions for multiple years?

MySQL Enterprise subscriptions have duration of at least 1 year. Customers have the flexibility of choosing terms with multi-year durations. To purchase multi-year contracts, please http://www.mysql.com/about/contact/sales.html?s=corporate

B.3: Can I buy MySQL Enterprise subscriptions for only some of my production MySQL database servers?

When you choose MySQL Enterprise subscriptions, they must cover all database servers that power that specific application. To negotiate volume discounts, please http://www.mysql.com/about/contact/sales.html?s=corporate

B.4: Do all my MySQL Enterprise subscriptions need to be at the same tier?

MySQL Enterprise subscriptions must be at the same tier (Basic, Silver, Gold, Platinum) for all database servers that power that specific application.

B.5: What if I plan to add more MySQL servers to my MySQL Enterprise subscription?

A great option is the MySQL Enterprise Unlimited offering that allows you cover an unlimited number of MySQL servers for a fixed, low price. http://www.mysql.com/products/enterprise/unlimited.html

B.6: How should I decide between MySQL Enterprise Basic, Silver, Gold and Platinum?

MySQL Enterprise subscriptions are available in 4 tiers, providing you the flexibility of choosing the capabilities and SLA that best meet your requirements. http://www.mysql.com/products/enterprise/features.html If you have questions and what to discuss your specific requirements, please http://www.mysql.com/about/contact/sales.html?s=corporate

B.7: What is MySQL Enterprise Server?

MySQL Enterprise Server software is the most reliable, secure and up-to-date version of MySQL for cost-effectively delivering E-commerce, Online Transaction Processing (OLTP), and multi-terabyte Data Warehousing applications. It is a fully integrated transaction-safe, ACID compliant database with full commit, rollback, crash recovery and row level locking capabilities. MySQL delivers the ease of use, scalability, and performance that has made it MySQL the world’s most popular open source database. http://www.mysql.com/products/enterprise/server.html

B.8: What is MySQL Production Support?

Production Support consists of 4 components:

- Problem Resolution Support
- Consultative Support
- Knowledge Base
- Technical Account Manager (option)

MySQL Production Support gives you priority access with guaranteed response times to assist you with the development, deployment and management of your MySQL applications. http://www.mysql.com/products/enterprise/support.html

B.9: Does MySQL Enterprise include 24x7 Technical Support?

MySQL Enterprise, at the Gold and Platinum tiers, includes 24x7 phone and email access to the MySQL Support Team. http://www.mysql.com/products/enterprise/problemresolution.html

B.10: Does MySQL Enterprise include Maintenance, Updates, and Upgrades?

Yes. As long as you have a valid contract for MySQL Enterprise, you will receive all new MySQL Enterprise Server software releases including Software Maintenance, Updates, and Upgrades. The Software Update Service will automatically notify you of the new releases.

B.11: Does MySQL Enterprise include Emergency Hot Fix Builds?

MySQL Enterprise, at the Gold and Platinum tiers, gives you the ability to request an Emergency Hot Fix Build to fix issues not already fixed in a MySQL Rapid Update or MySQL Quarterly Service Pack.

B.12: What is MySQL Consultative Support?

MySQL Enterprise, at the Gold and Platinum tiers, includes Consultative Support. This is a proactive approach to support that is designed to help you avoid critical outages. MySQL Support Engineers advise you on how to properly design and tune your MySQL servers, schema, queries, and replication set-up to maximize performance and availability. Also, by taking the initiative to properly design and tune your MySQL database applications you can avoid having to purchase expensive hardware for your IT infrastructure. http://www.mysql.com/products/enterprise/consultativesupport.html

B.13: What is a Technical Account Manager?

MySQL Enterprise, at the Platinum tier, provides the option for a Technical Account Manager (TAM). The TAM is your advocate within MySQL, who proactively works to maximize your benefits from MySQL Support Services. http://www.mysql.com/products/enterprise/tam.html

B.14: Does MySQL provide IP (Intellectual Property) Indemnification?

MySQL Enterprise, at the Gold and Platinum tiers, has the option of IP Indemnification, for qualifying customers at no extra cost. This provides you with legal protection that you expect from enterprise software providers. http://www.mysql.com/products/enterprise/indemnification.html

B.15: What is the list of Supported Platforms?

MySQL Enterprise provides broad coverage in its list of Supported Platforms. http://www.mysql.com/support/supportedplatforms/

B.16: Are there any Demo/Tutorials available for MySQL Enterprise?

Yes. Multiple self-running demos are available. http://www.mysql.com/products/enterprise/demo.html

B.17: Are there any MySQL Enterprise White Papers available?

Yes. Detailed architecture, technology, and business white papers are available. http://www.mysql.com/products/enterprise/whitepapers.html

B.18: Are there any Webinars available?

Yes. MySQL provides regularly scheduled Live Webinars. http://www.mysql.com/news-and-events/web-seminars/index.html MySQL also provides On-Demand Webinars to fit your schedule. These are recordings of previously held Live Webinars that you can replay at any time. http://www.mysql.com/news-and-events/web-seminars/index.html

B.19: What is the pricing of MySQL Enterprise?

The pricing model for MySQL Enterprise is based on two key components: per server and per year. MySQL Enterprise does not have artificial restrictions based on CPUs, Memory, Machine Size, or Named Users. MySQL Enterprise is available in 4 tiers (Basic, Silver, Gold and Platinum). Choose the tier that best meets your requirements and budget. http://www.mysql.com/products/enterprise/features.html

B.20: What is MySQL Enterprise Unlimited?

MySQL Enterprise Unlimited is a unique offering that allows you to deploy an unlimited number of MySQL Enterprise Servers for the price of a single CPU of Oracle Enterprise Edition. http://www.mysql.com/products/enterprise/unlimited.html

B.21: How do I get a 30-day trial on MySQL Enterprise?

You can experience the MySQL Enterprise Monitor for 30 days by registering to receive an email with login instructions. http://www.mysql.com/trials/enterprise

B.22: How do I buy MySQL Enterprise?

For pricing and to buy MySQL Enterprise, visit the Online Shop For volume discounts or for more information, please http://www.mysql.com/about/contact/sales.html?s=corporate

MySQL Monitor

Questions

- B.1: What is MySQL Enterprise Monitor?
- B.2: What MySQL Enterprise subscription levels include the MySQL Enterprise Monitor?
- B.3: What are the features and related benefits of the MySQL Enterprise Monitor?
- B.4: What are the immediate benefits of implementing the MySQL Enterprise Monitor?
- B.5: What are the long-term benefits of the MySQL Enterprise Monitor?
- B.6: How is the MySQL Enterprise Monitor installed and deployed?
- B.7: How is the Enterprise Monitor web application architected?

- B.8: What makes MySQL Enterprise unique?
- B.9: What versions of MySQL are supported by the MySQL Enterprise Monitor?
- B.10: What operating system platforms are supported by the MySQL Enterprise Monitor?
- B.11: How do I get the MySQL Enterprise Monitor?
- B.12: What are the MySQL Enterprise Advisors and Advisor Rules?
- B.13: Which Advisors and features are included under different MySQL Enterprise subscription levels?
- B.14: Which set of Enterprise Advisors, Advisor Rules and features are best for my use of MySQL?
- B.15: How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?

Questions and Answers

B.1: What is MySQL Enterprise Monitor?

Included as part of a MySQL Enterprise subscription, the MySQL Enterprise Monitor is a distributed, web-based application that helps customers reduce downtime, tighten security and increase throughput of their MySQL servers by telling them about problems in their database applications before they occur. It is downloadable from the Enterprise Customer web site and is deployed within the safety of the customer datacenter.http://www.mysql.com/products/enterprise/monitor.html

B.2: What MySQL Enterprise subscription levels include the MySQL Enterprise Monitor?

The Enterprise Monitor is available under MySQL Enterprise subscription levels Silver, Gold and Platinum. http://www.mysql.com/products/enterprise/features.html

B.3: What are the features and related benefits of the MySQL Enterprise Monitor?

The MySQL Enterprise Monitor is like having a "Virtual DBA Assistant" at your side to recommend best practices to eliminate security vulnerabilities, improve replication, and optimize performance. For the complete features and benefits, visit the http://www.mysql.com/products/enterprise/monitor-features.html.

B.4: What are the immediate benefits of implementing the MySQL Enterprise Monitor?

Often MySQL installations are implemented with default settings that may not be best suited for specific applications or usage patterns. The MySQL Advisors go to work immediately in these environments to identify potential problems and proactively notify and advise DBAs on key MySQL settings that can be tuned to improve availability, tighten security, and increase the throughput of their existing MySQL servers

B.5: What are the long-term benefits of the MySQL Enterprise Monitor?

Over time, the task of managing even medium-scale MySQL server farms becomes exponentially more complicated, especially as the load of users, connections, application queries, and objects on each MySQL server increases. The Enterprise Monitor continually monitors the dynamic security, performance, replication and schema relevant metrics of all MySQL servers, so as the number of MySQL continues to grow, DBAs are kept up to date on potential problems and proactive measures that can be implemented to ensure each server continues to operate at the highest levels of security, performance and reliability.

B.6: How is the MySQL Enterprise Monitor installed and deployed?

The Enterprise Monitor is powered by a distributed web application that is installed and deployed within the confines of the corporate firewall.

B.7: How is the Enterprise Monitor web application architected?

The Enterprise Monitor web application is comprised of 3 components:

- Monitor Agent: A lightweight C program that is installed on each of the monitored MySQL servers. Its purpose is to collect MySQL SQL and operating system metrics that allow the DBA to monitor the overall health, availability and performance of the MySQL server. The Monitor Agent is the only component within the application that touches or connects to the MySQL Server. It reports the data it collects using XML over HTTP to the centralized Service Manager.

- Service Manager: The main server of the application. The Service Manager manages and stores the data collections that come in from each monitor agent. It analyzes these collections using MySQL provided best practice Advisor rules to determine the health, security, availability and performance of each of the monitored MySQL Servers. The Service Manager also provides the content for the Enterprise Dashboard which serves as the client user interface for the distributed web application.
- Repository: A MySQL database that is used to stored data collections and application-level configuration data.

B.8: What makes MySQL Enterprise unique?

Of the products on the market that monitor MySQL, SQL code and OS specific metrics, the MySQL Enterprise Monitor is the only solution that is built and supported by the engineers at MySQL. Unlike other solutions that report on raw MySQL and OS level metrics, the MySQL Enterprise Monitor is designed to optimize the use of MySQL by proactively monitoring MySQL instances and providing notifications and “MySQL DBA expertise in a box” advice on corrective measures DBAs can take before problems occur.

B.9: What versions of MySQL are supported by the MySQL Enterprise Monitor?

The MySQL Enterprise Monitor can be used to monitor MySQL versions 4.0 – 5.x.

B.10: What operating system platforms are supported by the MySQL Enterprise Monitor?

The Enterprise Monitor Service Manager is fully supported on most current versions of Linux, Windows XP and Server Editions, Solaris and Mac OSX. The Monitor Agent supports any platform supported by the MySQL Enterprise server. For the complete list of MySQL Enterprise supported operating systems and CPUs, visit the http://www.mysql.com/support/supportedplatforms/enterprise.html.

B.11: How do I get the MySQL Enterprise Monitor?

The MySQL Enterprise Monitor is available for download to MySQL Enterprise customers at the Silver, Gold and Platinum subscription levels.

- To experience the MySQL Enterprise Monitor for 30 days, visit the http://www.mysql.com/trials/enterprise
- To buy MySQL Enterprise, visit the Online Shop

B.12: What are the MySQL Enterprise Advisors and Advisor Rules?

The MySQL Enterprise Advisors are a set of best practice guidelines for the optimal use of MySQL. Advisors are spread across database specific disciplines and are comprised of a set of MySQL Advisor Rules that proactively monitor all MySQL servers and report on database application problems before they occur. Each Advisor Rule provides a detailed overview of the problem it is designed to identify, advices on how to correct the problem, specifies commands to implement the recommended fix and links to resources for additional research into the issue at hand. http://www.mysql.com/products/enterprise/advisors.html

B.13: Which Advisors and features are included under different MySQL Enterprise subscription levels?

For the complete list of the MySQL Enterprise Advisors that are available under each MySQL Enterprise subscription level, visit the http://www.mysql.com/products/enterprise/features.html.

B.14: Which set of Enterprise Advisors, Advisor Rules and features are best for my use of MySQL?

The Enterprise Monitor Advisors and Advisor Rules are available at 3 MySQL Enterprise subscription tiers: Choose MySQL Enterprise Silver if you need:

- Assurance you are running the most current, bug-free version of MySQL across all of your servers.
- Recoverability of your MySQL servers.
- The highest level of security for your MySQL servers.
- Monitoring of maximum or disallowed MySQL connections.
- Optimized startup configuration settings.

Choose MySQL Enterprise Gold, when you need everything in Silver, PLUS:

- Easy collection and detection of problematic SQL code running on your production or development systems.
- Insight and corrective advice on MySQL replication status, sync, and performance related issues.
- Auto detection and documenting of your Replication topologies.
- Advanced monitoring of your Replication and Scale-out environment.

Choose MySQL Enterprise Platinum, when you need everything in Gold, PLUS:

- Identification and advice on unplanned database and object level schema changes (Create, Alter, and Drop) across your MySQL servers.
- Proactive monitoring and advice on tuning the performance of your MySQL servers.

B.15: How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?

Customers will receive notifications of new and updated MySQL Enterprise Monitor and Advisors as they become available through the MySQL Enterprise Software Update Service. Notifications will be generated and sent based on the customer profile and the MySQL Enterprise subscription level.

MySQL Query Analyzer

Questions

- B.1: What is the MySQL Query Analyzer?
- B.2: How is the MySQL Query Analyzer installed and enabled?
- B.3: What overhead can I expect when the MySQL Query Analyzer is installed and enabled?
- B.4: Can I leave the MySQL Query Analyzer enabled at all times?
- B.5: What are the main features and benefits of the MySQL Query Analyzer?
- B.6: What are the typical use cases of the MySQL Query Analyzer?
- B.7: How are subscribers notified about updates to the MySQL Query Analyzer application components?
- B.8: What makes the MySQL Query Analyzer unique?
- B.9: How can I get the MySQL Query Analyzer?
- B.10: Does Query Analyzer work with MySQL Cluster?
- B.11: Does Query Analyzer capture queries by the root user?
- B.12: Does Query Analyzer enable me to monitor the disk reads and writes during a query?
- B.13: Does Query Analyzer handler prepared statements?
- B.14: How much degration in performance does mysql-proxy introduce?
- B.15: Does the query analyzer look at all queries? or only queries which would show up in the in the slow-queries log?
- B.16: Does the "Rows" area show the rows returned/updated or the rows visited by the query?
- B.17: Do the MySQL clients have to connect to the port of the mysql proxy to enable the QA?

Questions and Answers

B.1: What is the MySQL Query Analyzer?

The MySQL Query Analyzer allows DBAs, developers and system administrators to improve application performance by collecting, monitoring, and analyzing queries as they run on their MySQL servers. http://www.mysql.com/products/enterprise/query.html

B.2: How is the MySQL Query Analyzer installed and enabled?

The Query Analyzer feature is installed with the Monitor Agent. It is enabled during agent installation and can be toggled between collection and pass-thru modes from the Query Analysis page of the Enterprise Monitor.

B.3: What overhead can I expect when the MySQL Query Analyzer is installed and enabled?

The average overhead when in active collection mode is in the 15-20% range. In pass-thru mode the overhead is minimal, weighing in at 1-5% on most MySQL systems of average load.

B.4: Can I leave the MySQL Query Analyzer enabled at all times?

We have customers who have the Query Analyzer enabled and collecting queries on their development and QA servers so they can tune their code and monitor the fixes as part of the development process. For production systems, Query collection and analysis can easily be toggled on when a slowdown occurs. To avoid collection mode overhead many users are using simple scripts to enable the Query Analyzer to sample queries during nonpeak hours, typically during 30 minute windows. They can then view the collected queries using the date/time or interval filter options.

B.5: What are the main features and benefits of the MySQL Query Analyzer?

For the complete features and benefits, visit the http://www.mysql.com/products/enterprise/monitor-features.html

B.6: What are the typical use cases of the MySQL Query Analyzer?

The typical use cases for developers, DBAs and system administrators are:

- Developers – Monitor and tune application queries during development before they are promoted to production.
- DBAs and System Administrators – Identify problem SQL code as it runs in production and advise development teams on how to tune. This use case benefits the most from regular sampling of queries as they are running, most often during nonpeak hours.

B.7: How are subscribers notified about updates to the MySQL Query Analyzer application components?

Customers will receive notifications of the MySQL Query Analyzer updates as they become available through the MySQL Enterprise Software Update and Alert Service. Notifications will be generated and sent based on the customer profile and the MySQL Enterprise subscription level.

B.8: What makes the MySQL Query Analyzer unique?

Other products (free, open source and commercial) that provide MySQL query monitoring are dependent on the MySQL Slow Query Log being enabled and available for sampling. While this provides some time savings over the DBA collecting and parsing the Log, the Slow Query Log comes with overhead and does not capture sub millisecond executions. The log data also grows very large very quickly.

The MySQL Query Analyzer collects queries and execution statistics with no dependence on the SQL Query Log, it captures all SQL statements sent to the MySQL server and provides an aggregated view into the most expensive queries in number of executions and total execution time. It is also fully supported as part of the MySQL Enterprise subscription.

B.9: How can I get the MySQL Query Analyzer?

The MySQL Query Analyzer is available for download to MySQL Enterprise customers at the Gold and Platinum subscription levels.

- To experience the MySQL Enterprise Monitor for 30 days, visit the http://www.mysql.com/trials/enterprise
- To buy MySQL Enterprise, visit the Online Shop

B.10: Does Query Analyzer work with MySQL Cluster?

Yes, providing that eact node is monitored with an agent and query analyzer has been unabled for that node. Note that you must be accessing your cluster data through a standard MySQL node for this to work.

B.11: Does Query Analyzer capture queries by the root user?

Yes, Query Analyzer captures all queries by all users providing that the queries are sent through the proxy port configured by the MySQL Enterprise Monitor Agent.

B.12: Does Query Analyzer enable me to monitor the disk reads and writes during a query?

No, that information is not available to be shown at this time.

B.13: Does Query Analyzer handler prepared statements?

At this time, the query analyzer does not track server-side prepared statements. However the default configurations for most client-side libraries for MySQL don't use them, they emulate them client-side, and those will be tracked by the query analyzer.

B.14: How much degration in performance does mysql-proxy introduce?

At the very least it's equivalent to a network hop in latency. The degredation is directly related to your average query execution time. If your queries execute in microseconds (which can happen if served from query cache) then the degradation will be higher, and noticable. We've seen some applications that actually do work when they execute queries, the degradation is much less, and in some limited cases because of scheduling, the application actually has better throughput.

B.15: Does the query analyzer look at all queries? or only queries which would show up in the in the slow-queries log?

The Query Analyzer sees all queries that you route through the agent/proxy that performs the query analysis and aggregate them directly.

B.16: Does the "Rows" area show the rows returned/updated or the rows visited by the query?

Returned/updated. We don't have visibility into how many rows were touched. at an instance level. Some of the graphs we provide will show you when you're queries are touching a lot of rows.

B.17: Do the MySQL clients have to connect to the port of the mysql proxy to enable the QA?

Yes, or you can re-direct them in various ways, by reconfiguring mysqld to listen to some other port, and the proxy to 3306, use iptables redirection, etc. We have some examples in the manual for the product on how to do it (semi)-transparently.

Appendix C. Files Associated with The MySQL Enterprise Monitor

> **Note**
>
> MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

This section details the files associated with the MySQL Enterprise Monitor and shows their default location.

C.1. Log Files for the MySQL Enterprise Service Manager

This section shows the location of the log and configuration files associated with the various components that make up the MySQL Enterprise Service Manager. These files can prove useful for debugging purposes.

Table C.1. MySQL Enterprise Monitor: Log File Locations: Windows

Component	File Location
Apache/Tomcat	C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\logs\catalina.out
Repository	C:\Program Files\MySQL\Enterprise\Monitor\mysql\data
Configuration Report	C:\Program Files\MySQL\Enterprise\Monitor\configuration_report.txt
Service Manager	C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\logs\mysql-monitor.log

Table C.2. MySQL Enterprise Monitor: Log File Locations: Unix

Component	File Location
Apache/Tomcat	/opt/mysql/enterprise/monitor/apache-tomcat/logs/catalina.out
Repository	/opt/mysql/enterprise/monitor/mysql/data
Configuration Report	/opt/mysql/enterprise/monitor/configuration_report.txt
Service Manager	/opt/mysql/enterprise/monitor/apache-tomcat/logs/mysql-monitor.log

Table C.3. MySQL Enterprise Monitor: Log File Locations: Mac OS X

Component	File Location
Apache/Tomcat	/Applications/mysql/enterprise/monitor/apache-tomcat/logs/catalina.out
Repository	/Applications/mysql/enterprise/monitor/mysql/data
Configuration Report	/Applications/mysql/enterprise/monitor/configuration_report.txt
Service Manager	/Applications/mysql/enterprise/monitor/apache-tomcat/logs/mysql-monitor.log

All of these logs are viewable within MySQL Enterprise Dashboard. For more information, see Section 4.6, “Logs”.

> **Note**
>
> The `configuration_report.txt` file contains the repository password in plain text. You may want to store this file in a secure location.

On all operating systems, the Apache/Tomcat, and Repository directories contain both access and error files.

C.2. Monitor Agent and Service Manager Installation Log Files

These log files are called `bitrock_installer.log`.

On Windows these files are stored in the temporary directory defined by the variable `%TEMP%`. To determine the value of `%TEMP%` on your system, type `echo %TEMP%` at the command line. Alternately, you may open the `Run` command window, enter `%TEMP%` and press OK. To find this file on Unix and Mac OS X check the value of the environment variable, `$TMPDIR`. If this variable is not defined, look in the `/tmp` directory.

If the file `bitrock_installer.log` already exists, a file called `bitrock_installer_xxxx.log` (where *`xxxx`* stands for an arbitrary series of numerals) will be created.

To determine if an installation file belongs to the Monitor Agent or the Service Manager you must open the file in a text editor.

C.3. Agent Log and PID Files

The locations of the agent log files are as follows:

- Windows: `C:\Program Files\MySQL\Enterprise\Agent\mysql-monitor-agent.log`
- Unix: `/opt/mysql/enterprise/agent/mysql-monitor-agent.log`
- Mac OS X: `/Applications/mysql/enterprise/agent/mysql-monitor-agent.log`

On any platform, the location of the agent log file may be changed. You may also change the name of the log file. To confirm the location and the name, check the settings in the `mysql-monitor-agent.ini` file.

The maximum size of a log file may be limited to 2GB. If MySQL Enterprise Monitor Agent is unable to add information to the configured logfile, then information will instead be sent to the standard output.

Because the size of the log files can become quite large, you can use a tool such as `logrotate` (Unix/Linux) or `logadm` (Solaris) to automatically rotate the logs. For example, a sample `logrotate` definition is shown below:

```
/opt/mysql/enterprise/agent/mysql-monitor-agent.log {
    rotate 5
    daily
    minsize 1G
    postrotate
        /usr/bin/killall -HUP mysql-monitor-agent
    endscript
}
```

The above will rotate the log before it reaches 1GB in size, keeping up to 5 files at a time.

If you want to use your own system, you can copy the log file, flush the contents, and then send the `SIGHUP` signal to the agent process. On Windows, you need to shutdown, copy the log, and restart the agent.

The locations of the agent `pid` files are as follows:

- Windows: `C:\Program Files\MySQL\Enterprise\Agent\agent\mysql-monitor-agent.pid`
- Unix: `/opt/mysql/enterprise/agent/mysql-monitor-agent.pid`
- Mac OS X: `/Applications/mysql/enterprise/agent/mysql-monitor-agent.pid`

On any platform, the location of the agent log file may be changed. You may also change the name of the log file. To confirm the location and the name, check the settings in the `mysql-monitor-agent.ini` file.

C.4. The Management Information Base (MIB) File

A MIB file is a requirement for using SNMP traps. A table showing the location of this file follows.

Table C.4. MySQL Enterprise Monitor: MIB File Locations

Operating System	File Location
Windows	C:\Program Files\MySQL\Enterprise\Monitor\support-files\MONITOR.MIB
Unix	/opt/mysql/enterprise/monitor/support-files/MONITOR.MIB
Mac OS X	/Applications/mysql/enterprise/monitor/support-files/MONITOR.MIB

C.5. The `config.properties` File

The repository user name and password are stored in in the `config.properties` file. A table showing the location of this file follows:

Table C.5. MySQL Enterprise Monitor: The `config.properties` File

Operating System	File Location
Windows	C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\webapps\ROOT\WEB-INF
Unix	/opt/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF
Mac OS X	/Applications/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF

You should make sure that the file is secured at the file system level so that it cannot be read by anybody but the administrator or Enterprise monitor.

Appendix D. Error codes

The following table documents error codes reported by the GUI and the associated message.

Table D.1. MySQL Enterprise Dashboard: GUI Error Codes and Messages

Error Code	Message
U0001	"*replaceable_value*" is a required field.
U0002	You must log in to access the requested resource.
U0003	You do not have permissions to access this resource.
U0004	The user field "*replaceable_value*" must not be null. Either the submission was broken or you have accessed this resource improperly.
U0005	A non-numeric value was submitted for the field "*replaceable_value*". Either the submission was broken or you have accessed this resource improperly.
U0007	Unable to connect to verify credentials. *replaceable_value*
U0008	Your credentials do not appear to be valid.
U0009	The uploaded Advisor jar was invalid.
U0010	The username and password combination was not found.
U0011	The user "*replaceable_value*" does not exist.
U0012	You may not log into the interface with agent credentials.
U0013	At least one agent is required to use the application.
U0014	At least one admin is required to use the application.
U0015	The uploaded product key is invalid.
U0016	You must provide a password for your Enterprise Credentials.
U0017	The Enterprise Credential passwords do not match.
U0018	You must provide an admin password.
U0019	The admin passwords do not match.
U0020	You must provide an agent password.
U0021	The agent passwords do not match.
U0022	You must provide Enterprise Credentials or upload a product key.
U0023	You must provide a non-zero interval.
U0024	"*replaceable_value*" is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0025	"*replaceable_value*" is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0026	Your "To" and "From" dates cannot be the same date.
U0027	You must provide a graph id for display.
U0028	You have exceeded the maximum number of graphs for display. You may only select *replaceable_value* graphs.
U0029	No graphs ids were found.
U0030	You must provide a non-zero interval.
U0031	Your full-sized width must be greater than or equal to *replaceable_value*.
U0032	"*replaceable_value*" is not a valid value for full-sized width.
U0033	Your full-sized height must be greater than or equal to *replaceable_value*.
U0034	"*replaceable_value*" is not a valid value for full-sized height.
U0035	Your thumbnail width must be greater than or equal to *replaceable_value*.
U0036	"*replaceable_value*" is not a valid value for thumbnail width.
U0037	Your thumbnail height must be greater than or equal to *replaceable_value*.
U0038	"*replaceable_value*" is not a valid value for thumbnail height.

Error Code	**Message**
U0039	You must select one or more events to close.
U0041	"*replaceable_value*" is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0042	"*replaceable_value*" is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0043	You must provide a non-zero interval.
U0044	Your width must be greater than or equal to *replaceable_value*.
U0045	"*replaceable_value*" is not a valid value for width.
U0046	Your height must be greater than or equal to *replaceable_value*.
U0047	"*replaceable_value*" is not a valid value for height.
U0048	You must provide a non-zero interval.
U0049	"*replaceable_value*" is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0050	"*replaceable_value*" is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0051	Your "To" and "From" dates cannot be the same date.
U0052	Your width must be greater than or equal to *replaceable_value*.
U0053	"*replaceable_value*" is not a valid value for width.
U0054	Your height must be greater than or equal to *replaceable_value*.
U0055	"*replaceable_value*" is not a valid value for height.
U0056	You must select one or more adivsors.
U0057	No schedule was found with id "*replaceable_value*".
U0058	You must select one or more rules to schedule.
U0059	There was a problem scheduling "*replaceable_value*" against "*replaceable_value*".
U0060	No rule was found with the uuid "*replaceable_value*".
U0061	Advisor name must not be empty.
U0062	Advisor "*replaceable_value*" already exists.
U0063	New advisor name must not be empty.
U0064	Advisor id must not be null.
U0065	No advisor was found with the id *replaceable_value*.
U0066	Rule uuid must not be null.
U0067	No rule was found with the uuid "*replaceable_value*".
U0068	One or more variables is required to make a functioning rule.
U0069	Each defined variable must be mapped to an instance. The variable "*replaceable_value*" is missing the instance to collect against.
U0070	Please choose an existing Advisor or create a new one with a non-empty name.
U0071	The default frequency must be a non-zero value.
U0072	You must enter a rule name.
U0073	You must have provided Enterprise credentials to use this feature. See the "MySQL Enterprise Credentials" section on the "Global Settings" page.
U0074	You must have provided Enterprise credentials to use this feature. Please contact someone with an Admin account to enter this information for you.
U0075	You must specify a file for upload.
U0077	You must provide an email address.
U0078	The provided email address appears to be invalid.
U0079	A valid SMTP server address must be given.
U0080	Your passwords do not match.
U0081	A valid SNMP target must be given.
U0082	A valid SNMP port must be given.

Error Code	Message
U0083	A valid SNMP community string must be given.
U0084	The given file could not be found.
U0085	The given file is not a valid license file.
U0086	An invalid file name was given.
U0087	The content of the given key file is invalid.
U0088	An invalid SNMP severity level was given.
U0090	An invalid data life span was given.
U0091	An invalid re-inventory frequency was given.
U0092	Passwords for MySQL Enterprise must not be empty.
U0093	The user name is missing.
U0094	An invalid user role was given.
U0095	An invalid time zone was given.
U0096	An invalid locale was given.
U0097	The group name is invalid.
U0098	The recipients list is empty.
U0099	Cannot find the e-mail target.
U0100	The given target id is invalid.
U0101	The group name must not be empty.
U0102	The server name must not be empty.
U0103	An invalid group id was given.
U0104	The group already exists.
U0105	This group name is already in use. Enter a different name.
U0106	Cannot find group with id `replaceable_value`.
U0107	Cannot find server with id `replaceable_value`.
U0108	An invalid server id was given.
U0109	Invalid Username
U0110	Invalid user role specified.
U0111	A user with the given name already exists.
U0112	The password must not be empty.
U0113	The user "`replaceable_value`" does not exist.
U0114	The log name must not be empty.
U0115	An invalid log level was given.
U0117	An invalid filter was set.
U0118	"`replaceable_value`" is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0119	"`replaceable_value`" is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0120	No event was found with the id "`replaceable_value`".
U0121	No category was found with the id "`replaceable_value`".
U0122	You must select one or more servers to add.
U0123	No log was found with the name "`replaceable_value`".
U0124	No schedule was found with the id "`replaceable_value`".
U0125	No notification group was found with the id "`replaceable_value`".
U0126	Unable to retrieve Advisors at this time.
U0127	Your MySQL Enterprise trial has expired.
U0128	You must specify a file for upload.
U0129	The uploaded Graph XML was invalid.

Error Code	Message
U0130	The uploaded Graph had an insufficient version number, and was not loaded.
U0131	No group was found with the id "replaceable_value".
U0132	No user was found with the name "replaceable_value".
U0133	No log was found with the key "replaceable_value".
U0134	An invalid e-mail address was given.
U0135	No user was found with the id "replaceable_value".
U0136	No graph was found with the id "replaceable_value".
U0137	You must provide a query summary id for display.
U0138	No query summary was found with the id "replaceable_value".
U0139	"replaceable_value" is not a valid UUID.
U0140	You are not authorized to access this resource.
U0141	No graph was found with the uuid "replaceable_value".
U0142	The target name is already in use.
U0143	No key was provided to sort on.
U0144	U0121 Unable to import subscription key. You may download the key manually from the Enterprise website and import it on this page.
U0145	U0120 Unable to import Advisor Jar. You may download the jar manually from the Enterprise website and import it on this page.
U0146	Unable to schedule rule "replaceable_value" due to "replaceable_value.replaceable_value" data not being collected from server "replaceable_value". It may be an unsupported collection for that server.
U0147	The requested feature, replaceable_value, has not been implemented.
U0148	You must select one or more servers.
U0149	You must provide a non-empty alias.
U0150	No server was found with the id "replaceable_value".
U0151	Server already exists with name "replaceable_value".
U0152	No server name provided
U0153	Your subscription has expired and you have exceeded your 30 day grace period.
U0154	Query Analyzer Data purge value cannot be greater than Historic Data purge value.
U0155	A MySQL Enterprise Trial product key has previously been installed. To subscribe to this service, contact replaceable_value, visit replaceable_value or call: &lt;blockquote&gt;&lt;table&gt;&lt;tr&gt;&lt;td&gt;replaceable_value&lt;/td&gt;&lt;td&gt;re- &lt;/td&gt;&lt;td&gt placeable_value&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;replaceable_value;replaceable_value&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/blockquote&gt;
U0156	The database cannot parse the regular expression: replaceable_value.
U0157	Query Explain support requires Query Example capture to be enabled.
U0158	This license has expired. To subscribe to this service, contact replaceable_value, visit replaceable_value or call: &lt;blockquote&gt;&lt;table&gt;&lt;tr&gt;&lt;td&gt;replaceable_value&lt;/td&gt;&lt;td&gt;re- &lt;/td&gt;&lt;td&gt placeable_value&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;replaceable_value;replaceable_value&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/blockquote&gt;
U0159	This Role does not have permission to configure Query Analyzer. Please contact an Administrator.
U0161	Please import a replaceable_value level Advisor .jar to use with this replaceable_value level product key.
U0162	The admin and agent usernames must be different.
U0163	The requested alias is too long; aliases must be less than replaceable_value characters.
U0164	"replaceable_value" is not a DNS resolvable hostname or IP address.

Error Code	Message
U0165	Heat chart rule "*replaceable_value*" scheduled against "*replaceable_value*" may not be unscheduled. This rule is required for the product to function properly.
U0166	Heat chart rule "*replaceable_value*" scheduled against "*replaceable_value*" may not be disabled. This rule is required for the product to function properly.
U0167	Bad graph identifier.
U0168	There is no graph with the given identifier.
U0169	Graph width "*replaceable_value*" is invalid.
U0170	Graph height "*replaceable_value*" is invalid.
U0171	Graph width "*replaceable_value*" is below minimum "*replaceable_value*".
U0172	Graph height "*replaceable_value*" is below minimum "*replaceable_value*".
U0173	Interval "*replaceable_value*" is not valid.
U0175	A HTTP proxy host is required; the port is optional.
U0176	Unable to parse the HTTP host "*replaceable_value*".
U0177	Unable to parse the HTTP proxy port number "*replaceable_value*".
U0178	An invalid hostname or port was given.
U0179	The HTTP proxy passwords must match.
U0180	You cannot change the role of the logged-in user.
U0181	No updates are available to install at this time.
U0182	There was an error formatting the Advisor Readme.
U0183	You must provide a primary hostname.
U0184	Invalid value for LDAP encryption.
U0185	Invalid value for LDAP referrals.
U0186	You must provide a login when using comparison mode.
U0187	You must provide a password attribute when using comparison mode.
U0188	You must provide a password digest when using comparison mode.
U0189	Unknown password digest provided.
U0190	LDAP passwords do not match.
U0191	User or role pattern required when using role mapping.
U0192	You must provide a role search pattern attribute.
U0193	You must provide at least one mapping for the "admin" role.
U0194	You must provide at least one mapping for the "dba" role.
U0195	You must provide at least one mapping for the "agent" role.
U0196	Disabling LDAP authentication for a user requires a new password to be created.
U0197	LDAP authentication can not be disabled when LDAP is configured to be authoritative.
U0198	The password can not be changed for users authenticated via LDAP.
U0199	The role can not be changed for users with LDAP mapped roles.
U0200	The user name can not be changed when LDAP is authoritative.
U0201	You must provide at least one mapping for the "read-only" role.
U0202	*replaceable_value* heat chart {0,choice,1#rule|1&lt;rules} against *replaceable_value* {1,choice,1#server|1&lt;servers} will not be unscheduled. {0,choice,1#This rule is|1&lt;These rules are} required for the product to function properly.
U0203	No schedules were found for the rule.
U0204	Delete without confirmation disallowed.
U0205	Nothing is selected.
U0206	No file was provided.
U0207	Unable to determine the type of this file.

Error Code	Message
U0208	Nothing is selected.
U0209	The selection is invalid.
U0210	A version (e.g., "1.0") is required.
U0211	*replaceable_value* heat chart {0,choice,1#rule|1&lt;rules} against *replaceable_value* {1,choice,1#server|1&lt;servers} will not be disabled. {0,choice,1#This rule is|1&lt;These rules are} required for the product to function properly.
U0212	You must specify a user search pattern or user search criteria.
U0213	You must choose user search by pattern or by criteria.
U0214	Script content file does not exist.
U0215	Script runtime does not exist.

The following table documents error codes reported by the Enterprise server and the associated message.

Table D.2. MySQL Enterprise Monitor: Server Codes and Messages

Error Code	Message
E0001	Internal Error: *replaceable_value*
E0100	Invalid user role "*replaceable_value*". Only "manager", "dba", and "agent" are valid.
E0101	Invalid user name "*replaceable_value*". User names must be alphanumeric and between 1-32 characters in length.
E0102	Invalid user password. Passwords must contain at least one number.
E0103	Can not delete root user.
E0104	User can not change role.
E0105	User "*replaceable_value*" already exists.
E0106	User "*replaceable_value*" does not exist.
E0107	User role "*replaceable_value*" is not authorized.
E0108	User "*replaceable_value*" has been previously deleted and is disabled.
E0200	No mapping exists to map the user "*replaceable_value*" to a MySQL Network user.
E0201	A mapping already exists for the user "*replaceable_value*" to a MySQL Network user.
E0202	MySQL Network is currently unavailable or could not be reached.
E0203	MySQL Network reported no alerts.
E0204	MySQL Network Advisor returned no data. Reason: "*replaceable_value*".
E0205	Your Advisors are currently up to date.
E0206	Your MySQL Enterprise trial expired after "*replaceable_value*".
E0207	The MySQL Enterprise subscription key is not valid.
E0208	No MySQL Enterprise subscription information is installed.
E0209	The MySQL Enterprise subscription key has expired.
E0210	User: *replaceable_value* Requires role: *replaceable_value*
E0211	You do not have permissions to access this resource.
E0212	The import of a duplicate MySQL Enterprise subscription trial key was rejected.
E0213	Unable to decode key from file: *replaceable_value*.
E0300	Can not delete default group.
E0301	Group "*replaceable_value*" already exists.
E0302	Group "*replaceable_value*" does not exist.
E0303	Invalid group name "*replaceable_value*". Group name must be alphanumeric and less than 128 characters.

Error Code	Message
E0304	Group element "*replaceable_value*" already exists.
E0305	Group element "*replaceable_value*" does not exist in the group "*replaceable_value*".
E0400	Agent "*replaceable_value*" does not exist.
E0401	Agent payload parameter NULL.
E0402	Agent task "*replaceable_value*" does not exist.
E0403	There has been a problem de-serializing this agent"s request. Please ensure all agents are version *replaceable_value* or greater.
E0500	Server "*replaceable_value*" already exists.
E0501	Server "*replaceable_value*" does not exist.
E0502	Server "*replaceable_value*" is actively being monitored and cannot be deleted. The last agent contact for this server was on *replaceable_value*.
E0503	Server "*replaceable_value*" is being migrated and cannot be deleted. Please halt the active migration first.
E0600	Database "*replaceable_value*" already exists.
E0601	Database "*replaceable_value*" does not exist.
E0700	Threshold "*replaceable_value*" already exists.
E0701	Threshold "*replaceable_value*" does not exist.
E0702	Data collection item does not exist.
E0703	serverName or groupName have to be set.
E0800	Data collection item does not exist.
E0801	Data collection is already occuring for "*replaceable_value*".
E0802	Data collection does not exist for "*replaceable_value*".
E0803	Invalid lifespan. Data life must be greater than 0 days.
E0804	Invalid namespace. Supported namespaces include "mysql" and "os". *replaceable_value*
E0805	Invalid instance. The instance must be a valid for data collection.
E0806	Invalid source. The source must be valid for data collection.
E0807	Invalid attribute. The attrib must be valid for data collection.
E0808	Invalid frequency: "*replaceable_value*". The frequency must be at least 1 minute and be in the format HH:MM.
E0809	Invalid port. If the URI uses the MYSQL namespace it must include a server port.
E0810	Invalid server. The server "*replaceable_value*" must exist in the current inventory.
E0811	Invalid server. For an OS namespace, the server must not include the MySQL server port.
E0812	A SQL failure occurred while saving the data collection schedule. "*replaceable_value*" Please see the error logs for details.
E0813	Invalid server id "*replaceable_value*".
E0814	No data items exist for server "*replaceable_value*".
E0815	Value "*replaceable_value*" not valid for "*replaceable_value*".
E0816	categoryName has to be set
E0817	You must specify either a server or group in the data collection schedule.
E0818	The server "*replaceable_value*" does not support item(s) "*replaceable_value*" required by monitor "*replaceable_value*".
E0819	Invalid data type: "*replaceable_value*".
E0900	Advisor id "*replaceable_value*" does not exist.
E0901	Advisor name "*replaceable_value*" does not exist.
E0902	Advisor XML is invalid.
E0903	Category "*replaceable_value*" does not exist.

Error Code	**Message**
E0904	Invalid server list.
E0905	SKIPPED "*replaceable_value*" for *replaceable_value* because Agent for *replaceable_value* is down.
E0906	SKIPPED "*replaceable_value*" for *replaceable_value* because mysqld on *replaceable_value* is down.
E0907	Category "*replaceable_value*" cannot be modified.
E0908	Category "*replaceable_value*" cannot be deleted.
E0909	Category "*replaceable_value*" already exists.
E0910	Advisor bundle version *replaceable_value* is incompatible with server version *replaceable_value*.
E0911	Heat chart rule "*replaceable_value*" scheduled against "*replaceable_value*" may not be unscheduled.
E0912	Heat chart rule "*replaceable_value*" scheduled against "*replaceable_value*" may not be disabled.
E1100	Graph time format "*replaceable_value*" is invalid. Must be in the format HH:MM:SS.
E1101	Graph size height and width must be specified.
E1102	Graph "*replaceable_value*" does not exist.
E1103	Graph "*replaceable_value*" already exists.
E1104	Graph "*replaceable_value*" contains a data collection item that can not be identified.
E1200	Failed to retrieve identity column.
E1201	Required parameter "*replaceable_value*" was NULL.
E1202	Only SELECT and SHOW commands are allowed via this interface.
E1203	Invalid timezone "*replaceable_value*".
E1204	Invalid interval "*replaceable_value*".
E1205	Could not find object "*replaceable_value*" in cache "*replaceable_value*".
E1206	Feature is not implemented. Parameters: *replaceable_value*
E1300	Advisor "*replaceable_value*" already exists.
E1301	Advisor "*replaceable_value*" does not exist.
E1302	Schedule does not exist. "*replaceable_value*" "*replaceable_value*"
E1303	JEP error: expression : "*replaceable_value*", message: "*replaceable_value*".
E1304	Advisor "*replaceable_value*" contains a data collection item that can not be identified.
E1305	Alarm level "*replaceable_value*" is not valid.
E1306	Schedule already exists. "*replaceable_value*" "*replaceable_value*".
E1307	Advisor "*replaceable_value*" is a read only MySQL Network certified Advisor. Please copy the rule before editing.
E1308	Advisor "*replaceable_value*" is currently scheduled against one or more monitored MySQL servers and cannot be removed.
E1309	Could not render text. "*replaceable_value*".
E1310	No open event for "*replaceable_value*" on server "*replaceable_value*". Perhaps already closed.
E1311	Alarm with id "*replaceable_value*" does not exist.
E1400	Invalid missed agent heartbeat threshold value.
E1401	Notification entry "*replaceable_value*" does not exist.
E1402	Agent "*replaceable_value*" is using session "*replaceable_value*" but session "*replaceable_value*" was requested.
E1403	Server name "*replaceable_value*" is in use by another agent with uuid "*replaceable_value*".
E1500	Notification target "*replaceable_value*" does not exist.
E1501	Invalid email address "*replaceable_value*".
E1502	Notification target "*replaceable_value*" already exists.

Error Code	Message
E1503	Invalid notification target name "*replaceable_value*".
E1504	Invalid monitor name, it must not be empty.
E1505	No variable assignments given, you must define at least one.
E1506	SMTP authentication failed.
E1507	SMTP send failed
E1508	Invalid SNMP target "*replaceable_value*".
E1509	Invalid SNMP trap type "*replaceable_value*".
E1600	Log "*replaceable_value*" does not exist.
E1700	Authentication Mechanism was null
E1701	Unsupported Authentication Mechanism *replaceable_value*
E1702	Incorrect password for user *replaceable_value*
E1800	Invalid data type: "*replaceable_value*", new value: "*replaceable_value*".
E1801	Invalid value: "*replaceable_value*" for data type " *replaceable_value* " for item "*replaceable_value*".
E1802	Invalid expression: "*replaceable_value*", exception: " *replaceable_value*", raw expression: " *replaceable_value*". substitute values: "*replaceable_value*".
E1900	History not found id: "*replaceable_value*", schedule: "*replaceable_value*", expression: "*replaceable_value*".
E2000	Tag not found, id: "*replaceable_value*".
E2101	Unable to collect "*replaceable_value*" on "*replaceable_value*" for instance "*replaceable_value*".
E9000	*replaceable_value*
E9001	*replaceable_value* from: *replaceable_value*

Appendix E. MySQL Enterprise Monitor Change History

This appendix lists the changes to the MySQL Enterprise Monitor, beginning with the most recent release. Each release section covers added or changed functionality, bug fixes, and known issues, if applicable. All bug fixes are referenced by bug number and include a link to the bug database. Bugs are listed in order of resolution. To find a bug quickly, search by bug number.

E.1. Changes in MySQL Enterprise Monitor 2.0.7 (Not yet released)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.6.

Bugs fixed:

- The agent did not start because the file `share/mysql-proxy/items/custom.xml` was missing from the build, but referred to in the configuration file. The log recorded the following error:

  ```
   Default log:
   ------------
   2009-09-18 16:25:12: (critical) chassis.c:1097: could not raise RLIMIT_NOFILE to 8192,
  Invalid argument (22). Current limit still 1024.
   2009-09-18 16:25:13: (critical) job_collect_mysql.c.1894: opening
  '/data0/merlin/agent/2.0.7.7163/openSUSE10.3-x86-64bit/qa-merlin/share/mysql-proxy/items/c
  ustom.xml' failed with: Failed to open file
  '/data0/merlin/agent/2.0.7.7163/openSUSE10.3-x86-64bit/qa-merlin/share/mysql-proxy/items/c
  ustom.xml': No such file or directory
   2009-09-18 16:25:13: ((error)) scheduler.c:318: register_classes() for plugin
  collect_mysql failed
  ```

  If the reference to `share/mysql-proxy/items/custom.xml` was removed from the “agent-item-files = ...” entry in the configuration file, then the agent started as expected. (Bug#47443)

- After upgrading to Mac OS X Snow Leopard, the following error was generated when starting an agent:

  ```
  Macintosh-6:replication ligaya$
  ./Master.5.1.34sp1/MEM_agent/etc/init.d/mysql-monitor-agent start
  Starting MySQL Enterprise agent service...
  ./Master.5.1.34sp1/MEM_agent/etc/init.d/mysql-monitor-agent: line 175:   861 Trace/BPT
  trap              nohup $MERLIN_AGENT_ROOTDIR/bin/mysql-monitor-agent $MERLIN_AGENT_OPTIONS >
  /dev/null
   ERROR! MySQL Enterprise agent failed to start.
  ```

  Further, when attempting to display the agent version the following error was generated:

  ```
  Macintosh-6:replication ligaya$ ./Master.5.1.34sp1/MEM_agent/bin/mysql-monitor-agent -v
  dyld: Library not loaded: /usr/lib/libxml2.2.dylib
    Referenced from:
  /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/DictionaryServices
  .framework/Versions/A/DictionaryServices
    Reason: Incompatible library version: DictionaryServices requires version 10.0.0 or
  later, but libxml2.2.dylib provides version 9.0.0
  Trace/BPT trap
  ```

  (Bug#47042)

E.2. Changes in MySQL Enterprise Monitor 2.0.6 (27 August 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.5.

Functionality added or changed:

- Events used a GMT timestamp for the event time within email notifications, rather than the local time for the server generating the event.

  MySQL Enterprise Monitor has been changed so that it displays the time using the server's locale and GMT. (Bug#43739)

Bugs fixed:

- Initial setup and registration of MySQL Enterprise Monitor could fail on Mac OS X when communicating with the MySQL Enterprise website. (Bug#46571)
- If a migration from 1.3 to 2.0 was initiated and a new server created during the migration, but then deleted before migration was finished, the new server delete operation failed with an Null Pointer Exception (NPE). (Bug#44991)
- The Rule “InnoDB Buffer Cache Hit Rate Not Optimal” in the “Memory Usage” Advisor did not contain an uptime check. This resulted in premature firing of an info event. (Bug#44770)
- In the Advisor “Administration” the Rule “InnoDB Redo Logs Not Sized Correctly” did not fire correctly. The rule contained the expression:

  ```
  (%have_innodb% == "YES") && ((%innodb_log_file_size% * %innodb_log_files_in_group%) <
  LEAST(1073741824, (%innodb_buffer_pool_size% / 2)))
  ```

  This was incorrect and needed to be changed to:

  ```
  (%have_innodb% == "YES") && ((%innodb_log_file_size% * %innodb_log_files_in_group%) <=
  LEAST(1073741824, (%innodb_buffer_pool_size% / 2)))
  ```

  (Bug#44734)
- The recommendation for the rule `Table Cache Not Optimal` says:

  ```
  Recommended Action
  SET GLOBAL table_cache = (64 + 16);
  ```

  But an error was generated on executing that query:

  ```
  mysql> SET GLOBAL table_cache = (64 + 16);
  ERROR 1193 (HY000): Unknown system variable 'table_cache'

  mysql> select version();
  +------------+
  | version()  |
  +------------+
  | 5.1.31-log |
  +------------+
  1 row in set (0.01 sec)
  ```

  (Bug#44602)
- When running the shutdown script for Enterprise Dashboard, it took an unacceptably long time for Tomcat to terminate completely, the `java` process eventually had to be killed manually. (Bug#44327)
- The upgrade installer overwrote any custom settings stored in `WEB-INF/config.properties`. For example it set the database host back to `localhost`. (Bug#44003)
- When updating an Agent from 2.0 to 2.1 the `mysql-monitor-agent.ini` was incorrectly updated.

  This happened if the Agent was configured to use SSL to connect to the Enterprise Dashboard, on a port other than 18443. The update installer caused any value specified for the port to be changed to 18443. This did not happen if the Agent was not using SSL. (Bug#43900)
- An error was generated when an attempt was made to rename a Replication Group to one that had previously been deleted. (Bug#43846)
- A new topology was not discovered after the previous replication group was renamed. (Bug#43815)
- On Unix systems, executing the command:

  ```
  ./mysqlmonitorctl.sh stop
  ```

  did not make sure that `mysqld` was shutdown before finishing.

This resulted in a situation such as the following:

```
# /opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh stop
Using CATALINA_BASE:   /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_HOME:   /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_TMPDIR: /opt/mysql/enterprise/monitor/apache-tomcat/temp
Using JRE_HOME:       /opt/mysql/enterprise/monitor-2.0.0.7092/java
Stopping tomcat service ... [ OK ]
/opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh : mysql stopped
```

However, running the following command a few minutes later showed that the MySQL server was still running:

```
# /opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh status
MySQL Network MySQL is running
MySQL Network Tomcat is not running
```

(Bug#43803)

- The MySQL Enterprise Monitor upgrade installer incorrectly replaced the `AdvisorScript.jar` in `<instDir>/apache-tomcat/webapps/ROOT/WEB-INF/lib/` with the default Advisor JAR. (Bug#43773)

- The agent created the `mysql.inventory` table with an engine type of InnoDB, instead of MyISAM, when InnoDB was specified as the default engine type in `my.cnf`. This happened because the agent did not explicitly specify the table engine type to be of MyISAM. (Bug#43551)

- The MySQL Enterprise Monitor installer failed to install correctly due to insufficient disk space, even though the installer calculated there was suffcient disk space for the installation. This was due to the installer having out of date information regarding disk space requirements. (Bug#43538)

- After removing enough servers to bring the host count back down to the number covered in the subscription, the Subscription Warnings reflected the new number. However, the warning message displayed:

  ```
  You are currently monitoring 1 host, however your subscription covers only 1. Your
  subscription needs to be updated to cover at least 0 additional hosts.
  ```

  Instead of displaying:

  ```
  Your subscription is up-to-date
  You have no warnings at this time.
  ```

  The former message resulted in confusion. (Bug#43163)

- If a host was not a slave during the initial discovery phase, then it would not be displayed in the Replication tab if it subsequently became a slave.

  This was because after the initial discovery phase, if a host did not have `slavestatus` present, no subsequent checks were made to check for the host being a slave. It was therefore missed for the purposes of replication discovery and never showed in the Replication tab. (Bug#42997)

- The advisor "User Has Rights To Database That Does Not Exist" generated erroneous alerts.

  If a database was created with an "_" character in the name, and then user privileges granted to this database using the escaped character sequence "_" to prevent wildcards, then the advisor generates an error stating there is no database for the privilege.

  For example, if the following is carried out on the monitored server:

  ```
  CREATE DATABASE test_foo;
  GRANT SELECT ON `test\_foo`.* to testuser@'localhost' identified by 'test';
  ```

  then the advisor warns that these users have rights to a database that does not exist:

  ```
  ''@'%' on DB test_%,
  'test'@'localhost' on DB test_foo,
  'testuser'@'localhost' on DB test_foo
  ```

  (Bug#41717)

- The generic Linux IA64 glibc2.3 Agent installer was missing from the build. (Bug#41224)
- The Enterprise Dashboard displayed a blank entry for **Disk Space** in the Meta Info area. This happened on Open Solaris 2008.05. This problem only occurred when using the ZFS file system. (Bug#40907)
- The graphs for Thread Cache, Connections and Temporary Tables contained incorrect Japanese translations on their Y axis. The Japanese displayed “total connection time (min)” when it should have displayed something else. For example, the Thread Cache graph should have displayed “total/min”. (Bug#40413)
- When the Agent was started as a service on Windows for the first time, the name in the Task Manager window was `MYSQL-~1.EXE`. This occurred whether the Agent was started from within the installer or from the Start Menu.

  If the service was restarted, the Agent's name changed to the correct value, `mysql-service-agent.exe`. (Bug#30166)

E.3. Changes in MySQL Enterprise Monitor 2.0.5 (18 March 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.4.

Bugs fixed:

- The FreeBSD 7 Agent was inadvertently a Linux binary.

  ```
  shell> file mysqlmonitoragent-2.0.5.7153-freebsd7-x86-32bit-installer.bin
  mysqlmonitoragent-2.0.5.7153-freebsd7-x86-32bit-installer.bin: ELF 32-bit LSB executable,
  Intel 80386, version 1 (SYSV), for GNU/Linux 2.2.5, dynamically linked (uses shared libs),
  stripped
  ```

  (Bug#44263)
- Calling the Agent with the option `--agent-run-os-tests` resulted in a crash. This happened on Linux x86-64 systems. The resultant stack trace was:

  ```
  (qa-merlin)
  2009-03-04 16:39:42: (critical) chassis.c:1097: could not raise RLIMIT_NOFILE to 8192,
  Invalid argument (22). Current limit still 1024.
  sigar-test-all.c.124 (test_sigar_pid_get):
    pid = 5188
  sigar-test-all.c.106 (test_sigar_mem_get):
  ...cut...
  sigar-test-all.c.427 (test_sigar_file_system_list_get): (items = 13)
  [0]
    fs.dirname = /
    fs.devname = /dev/mapper/vg00-root
    fs.typename = local
    fs.sys-type-name = ext3
    fs.type = 2
    fsusage.total = 15481840
    fsusage.free = 14116140
    fsusage.used = 1365700
    fsusage.avail = 13329708
    fsusage.files = 1966080
    fsusage.use_percent = 0.100000
  [0]
    diskusage.reads = 315302
    diskusage.writes = 6318240
    diskusage.write_bytes = 25879511040
    diskusage.read_bytes = 6561092608
    diskusage.queue = 47457530080206
  Segmentation fault
  ```

  On some systems no output was shown, other than the message “Segmentation fault”. (Bug#43381)
- Following a change in the replication configuration, MySQL Enterprise Monitor did not display the new topology correctly. (Bug#43240)
- When a data collection became invalid, the agent sent NULLs for those collection values. However, the timestamps that it sent with the values were the timestamps from the last valid value that was collected.

  Due to key constraints on the Service Manager side, MySQL Enterprise Monitor disregarded anything sent with duplicate timestamps, thus the NULLs received from the agent were never processed. Some mechanisms, such as replication discovery, depend on the NULLs to identify a situation where data has become invalid. (Bug#43239)

- MySQL Enterprise Monitor did not add a log entry each time data was purged. The log entry should have noted how many rows of each type of data were purged (historical data, logs, quan data). (Bug#43159)

- MySQL Enterprise Monitor generated a stack overflow. This was as a result of a bug in Hibernate, which caused Hibernate to enter infinite recursion while trying to load a relationship. (Bug#43107)

- The “Table Cache Set Too Low For Startup” and “Table Cache Not Optimal” rules were not supported on MySQL 5.1 because the `table_cache` system variable was deprecated and replaced with `table_open_cache`. (Bug#42663)

- Migrated server was not completely deleted.

  In a Monitor that had been updated from 1.3.2 to 2.0.4, with 2 database servers queued for migration, if a server being migrated was deleted, or a migrated server was deleted, this would not be reflected in the user interface or in the license count, until Tomcat was restarted. (Bug#42604)

- The installer used to upgrade from version 1.3 corrupted passwords containing the “?” character. (Bug#42452)

- The replication status of a server was displayed as “unknown”. However, the agent logged no errors and the server showed no other event problems. The server status was displayed as “up” and the database connection was also displayed as “up”. (Bug#42407)

- Sun multi-core processors caused all cores to be reported on the meta information page.

  The larger T-series SPARC processors have 32+ cores. This caused the meta information page in the Dashboard to scroll as it reported each one. (Bug#42355)

- If an attempt was made to disable a rule using the link next to the rule, the following error message was generated:

  ```
  U0002 You must log in to access the requested resource. Go to login page.
  ```

  However, clicking the link did not prompt the user to login again. (Bug#42313)

- Changing `ssh-agent` from OpenSSH or specifying a malevolent value of `agent-host-id`, could inject data into the monitored MySQL Server.

  For example, setting `agent-host-id` to the value “I'm a test” would result in the following message in the error log:

  ```
  2009-01-23 15:45:11: ((error)) agent_mysqld.c:281: mysql_real_query('INSERT INTO
  mysql.inventory  (name, value) VALUES ( 'hostid', 'I'm a test' )') on 'mysql' failed: You
  have an error in your SQL syntax; check the manual that corresponds to your MySQL server
  version for the right syntax to use near 'm a test' )' at line 1 (mysql-errno = 1064)
  ```

  (Bug#42306)

- When `SHOW GLOBAL STATUS` returned a value greater than 214748364, it was sent to the Service Manager as 214748364. (Bug#42245)

- The Agent failed to identify local sockets as local on Mac OS X 10.4.

  If the Agent was configured to use a Unix domain socket on Mac OS X 10.4, it did not treat the connection as local and failed to provide CPU and memory information to MySQL Enterprise Monitor. This is shown in the log file:

  ```
  2009-01-20 18:15:02: (message) network-socket.c:752: is-local family 0 != 1
  2009-01-20 18:15:02: (message) agent_mysqld.c:322: [mysql] mysqld is not local or not
  directly connected
  ```

  However, this problem did not happen on Mac OS X 10.5. (Bug#42220)

- Some graphs on the **Graph** tab were not updated after the page was refreshed, or Update was clicked.

  The only way to get an updated graph was to change the graph size (in pixels) and then click Update. (Bug#42162)

- The `my.cnf` file for the Enterprise Monitor internal database had the following configuration item:

  ```
  innodb_autoextend_increment = 50M
  ```

This generated the error:

```
16:36:23 [Warning] option 'innodb_autoextend_increment': unsigned value 52428800
adjusted to 1000
```

This variable is interpreted as being specified in MB, so 50M would be 50 TB. Such a high value results in the variable being adjusted to 1000 MB.

The value in the configuration file should be:

```
innodb_autoextend_increment = 50
```

(Bug#42096)

- A number of Advisor rules had advice text that had not been translated into Japanese. The Advisors that contained untranslated rules included Performance, Schema and Security. (Bug#42067)
- The Service Manager did not handle the case where the agent failed to supply a valid `master_ip`. The Service Manager would then not restart. The logs contained the following:

  ```
  2009-01-09 14:39:50,472 ERROR [main:org.springframework.web.context.ContextLoader] Context
  initialization failed
  org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with
  name 'serverConnectionMonitor' defined in ServletContext resource
  [/WEB-INF/applicationContext.xml]: Unsatisfied dependency expressed through constructor
  argument with index 2 of type [com.mysql.etools.monitor.bo.Manager]: Error creating bean
  with name 'manager' defined in ServletContext resource [/WEB-INF/applicationContext.xml]:
  Invocation of init method failed; nested exception is
  com.mysql.etools.monitor.pom.UnsupportedAttributeException:
  101c6b5b-15eb-49aa-916c-843c51b28d38: mysql.slavestatus.Master_ip
  ```

  (Bug#42005)
- Having too many users with strong privileges generated Service Manager errors and events failed to be triggered.

  If there were approximately 1000 users with full privileges and the value of `group_concat_max_len` was set to 100001, the size of the data that the agent sent to the Service Manager was too large and caused Service Manager errors. Also, the Security event “Account has Strong MySQL privileges” did not trigger. (Bug#41987)
- Query Analyzer's Query Type filter for `SELECT` ignored statements starting with parenthesis.

  If you sent a statement through Query Analyzer starting with parenthesis, such as:

  ```
  (SELECT 1 FROM dual);
  ```

  and then attempted to filter for `SELECT` queries only, queries starting with parenthesis were not displayed. (Bug#41968)
- The agent angel process was spawning too soon, which caused a duplicate UUID error.

  `g_error()` logged a fatal error, and called `abort()`. This terminated the program. Then the angel respawned with the same UUID, but with a -1 session resync request, which the MEM server denied because the old session was still active. This resulted in a permissions denied error (1142) on the `mysql.inventory table`.

  ```
  2008-12-17 18:58:58: (message) agent_mysqld.c:313: [mysql] mysqld is local and directly connected
  2008-12-17 18:58:58: ((error)) agent_mysqld.c:444: [mysql] mysql_real_query("SELECT value
  FROM mysql.inventory WHERE name = 'uuid'") failed: SELECT command denied to user
  'ent_agent'@'127.0.0.1' for table 'inventory' (errno=1142)
  2008-12-17 18:58:58: (debug) chassis.c:282: 15134 returned: 15134
  2008-12-17 18:58:58: (message) chassis.c:304: [angel] PID=15134 died on signal=6 (it used
  0 kBytes max) ... waiting 3min before restart
  2008-12-17 18:59:00: (debug) chassis.c:244: we are the child: 15149
  2008-12-17 18:59:00: (message) chassis.c:259: [angel] we try to keep PID=15149 alive
  2008-12-17 18:59:00: (debug) chassis.c:277: waiting for 15149
  2008-12-17 18:59:00: (message) mysql-proxy 0.7.0 started
  2008-12-17 18:59:00: (message) MySQL Monitor Agent 2.0.0.7111 started.
  ```

  (Bug#41600)

- master_uuid discovery did not work with MySQL Server versions prior to 5.1. master_uuid did not show in any Agent to Monitor communications, and no log or error messages were generated.

  However, now the bug has been fixed, an Agent monitoring a 5.0 MySQL Server would register the following in its logs:

  ```
  ...
   <classname>slavestatus</classname>
   <instance>12515cdc-8c00-4223-9d2a-2666a403512c</instance>
   <attribute>Master_uuid</attribute>
  </target>
  <utc>2009-03-03T19:58:05.700Z</utc>
  <value>b2fd9f86-6e42-49f2-b930-e8fb3e728179</value>
  ```

  Note the presence of master_uuid. (Bug#41525)

- The master_uuid was not used for replication topology discovery.

  The agent collected master_uuid by reading the master.info file, and then running a SELECT directly against its master. This was to try and read the mysql.inventory table on the master to obtain the instance master_uuid.

  However, this was not being used correctly by the replication topology discovery within the server. (Bug#41519)

- Queries such as SELECT against the master mysql.inventory was not logged on slave-based agents. This made diagnosing topology discovery issues difficult. (Bug#41518)

- After an error was generated due to an incorrect password while trying to create a new user, the following error was obtained when subsequently attempting to create a valid new user:

  ```
  U0002 You must log in to access the requested resource
  ```

  (Bug#41384)

- Agent startup failed on Solaris 10. The error generated was:

  ```
  # ./mysql-monitor-agent start
  Bad string
  ERROR! /opt/mysql/enterprise/agent/etc/mysql-monitor-agent.ini has to have a
  [mysql-proxy].pid-file setting
  ```

  This was caused by unexpected behavior of the tr command. (Bug#41260)

- On the Query Analysis page, if a query was clicked the minimum displayed was greater than the average. (Bug#41259)

- In some circumstances the agent/proxy ran out of file descriptors, causing secondary failures. It could not recover from that state. The relevant part of the log file is shown here:

  ```
  2008-11-27 11:11:00: (critical) last message repeated 2 times
  2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
  2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
  2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
  2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
  2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
  2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
  2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
  2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
  2008-11-27 11:11:30: (critical) network-socket.c.292: socket(127.0.0.1:3306) failed: Too many open files (24)
  2008-11-27 11:11:30: (critical) proxy-plugin.c.1532: Cannot connect, all backends are down.
  2008-11-27 11:20:22: (critical) last message repeated 4 times
  2008-11-27 11:20:22: (critical) network-io.c:215:
  curl_easy_perform('https://user:password@merlin-dashboard:443/heartbeat') failed:
  SSL connection timeout (curl-error = 'Timeout was reached' (28), os-error = 'Connection refused' (111))
  ```

  (Bug#41068)

- If an installation of Service Manager 2.0.0.7102 included a backup directory, due to a previous upgrade, and was upgraded using at least Service Manager 2.0.0.7103, then the installer displayed an error message and exited.

  The error message displayed was:

  ```
  There has been an error.
  Error renaming /Applications/mysql/enterprise/monitor/apache-tomcat to
  ```

```
/Applications/mysql/enterprise/monitor/backup/apache-tomcat
The application will exit now
```

(Bug#40996)

- The Agent started without problems and connected to the master. But it was unable to perform a `SELECT` from the table `mysql.inventory` to obtain server information. (Bug#40933)

- Canonical Query Text for `Select -1` was displayed as `SELECT -?` instead of `SELECT ?` on the Query Analyzer tab. (Bug#40435)

- The agent installer for Solaris 8 x86 32-bit was missing. (Bug#40248)

- The MySQL Enterprise Monitor Dashboard did not display the subscribed notification group(s) on the Advisor **Current Scheduled** and **Add to Schedule** pages. (Bug#36007)

- The startup scripts supplied with MySQL Network Monitoring and Advisory tool did not supply all of the LSB `init.d` script options required. A list of the required options can be found at the following website http://refspecs.freestandards.org/LSB_3.1.0/LSB-Core-generic/LSB-Core-generic/iniscrptact.html

  The required options missing include `status` and `force-reload`. The option `status` is used by monitoring tools and cluster software such as Red Hat Cluster, to ensure that the service is still running. The `force-reload` option is an alias for restart. (Bug#29848)

- Multiple errors showed in the agent log after issuing a `SHOW INNODB STATUS` statement. The `InnoDB Buffer Pool`graph also went blank. (Bug#27372)

E.4. Changes in MySQL Enterprise Monitor 2.0.4 (05 February 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.3.

Bugs fixed:

- The Monitor Agent did not log connections to, and disconnects from, the monitored database.

  This meant it was not possible to check if the Agent was connected to the database by simply looking at the log file. (Bug#42403)

- The Monitor Agent failed to create the `mysql.inventory` table. The logs displayed the following error message:

  ```
  (critical) (share/mysql-proxy/quan.lua:711) [proxy] please add SELECT permissions for
  this user on mysql.inventory to enable the QUAN feature, got Table 'mysql.inventory'
  doesn't exist
  ```

  This happened even though the Monitor Agent used the `root` account. (Bug#42389)

- After installing Monitor Agent 2.0.4.7138 and then starting it with `--version`, the incorrect version was displayed. Although 2.0.4.7138 was installed, the Monitor Agent displayed the version number as 2.0.2.7138. (Bug#42263)

- An invalid path was shown in the error message if the upgrade installer failed to find the previous install location.

  The error message is shown below, note that the error message displays a different path to that provided by the user:

  ```
  Please specify the directory that contains the previous installation of the MySQL
  Enterprise Monitor Agent

  Installation directory [/home/mysql/mysql/enterprise/agent]: /var/lib/mysql/agent

  Warning: The directory /home/mysql/mysql/enterprise/agent does not contain a
  previous installation. Please select the right installation directory.
  Press [Enter] to continue :
  ----------------------------------------------------------------------------
  Please specify the directory that contains the previous installation of the MySQL
  Enterprise Monitor Agent
  ```

  (Bug#42200)

- The agent password error in the Service Manager did not log the originating host, making it impossible to determine the agent that failed to log in:

  ```
  ErrorJan 5, 2009 4:56:08 PM<?xml version='1.0'?><exceptions><error><![CDATA[E1702:
  IncorrectPasswordException: [agent]]]></error></exceptions>
  ```

  (Bug#42139)

- The Service Manager's JDBC connections did not have `sql_mode` set to include `NO_ENGINE_SUBSTITUTION`. This resulted in the failure of Data Definition Language (DDL) if the Service Manager was inadvertently pointed to an incorrect `mysqld` instance. (Bug#42137)

- A number of Advisor rules had advice text that had not been translated into Japanese. The Advisors that contained untranslated rules included Performance, Schema and Security. (Bug#42067)

- Monitor Agent configuration files had global read privileges on the filesystem. (Bug#41794)

- When the log files rotated to the maximum permitted by the `log4j` configuration, the metadata contained in the `FileAppender` became out of synchronization due to a logic bug. This caused an assertion error on any subsequent request for the logs tab or data. (Bug#41593)

- SNMP trap messages were sending 127.0.0.1 as the IP address, and there was no feature to enable the user to configure the IP address contained in the SNMP message, which would have been useful for troubleshooting. (Bug#41361)

- On all available test systems, including SUSE Linux Enterprise Server (SLES) 9 and 10 x86-32, x86-64, IA64, and Ubuntu 6.10 x86-64, the agent memory grew incrementally within ~3 hours to ~25M.

  The agents then switched to the 'red' condition on the dashboard and no more data was reported. The agent processes were still active.

  This was also present in MySQL Enterprise Monitor version 2.0.0.7111 on Suse 7.3, with the glibc2.2 x86 agent.

  There was no load on the monitored servers, and the problem also occurred when the “self-quan” was not configured. (Bug#41244)

- Although the **Monitor** tab loaded initially, after a 64-bit MySQL server running a 32-bit MySQL Monitor Agent was clicked, a Null Pointer Exception was generated. (Bug#41164)

- The Monitor Agent startup configuration did not seem to work and did not generate a log file. (Bug#40583)

- An error generated by a rule failed to clear.

  When a rule was created with the `os:disk:fs_used` data item, if the instance was not a valid mount point then the Monitor Agent reported the error:

  ```
  2008-08-11 17:57:00: (critical) disk-get failed for all: 2
  ```

  Note the above instance was for “all”, and similar results occurred for instance “disk”.

  The issue was that upon editing the rule, or even deleting the rule, the agent still showed the above error type. Restarting the agent and the monitoring service failed to remove the error. (Bug#38709)

- If the “On Save send test trap” checkbox was checked when the Save button was clicked and the locale was set to Japanese, an error occurred. The orange error banner was displayed at the top of the page with the error message in Japanese. (Bug#32069)

E.5. Changes in MySQL Enterprise Monitor 2.0.3 (23 January 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.2.

Bugs fixed:

- The Monitor Agent failed to connect to the server, and no error message was displayed in the log file.

  However, when the log verbosity level was set to `message`, the following message was recorded:

```
2009-01-12 13:34:43: (warning) agent_mysqld.c:600: agent connecting to mysql-server
failed: mysql_real_connect(host = '127.0.0.1', port = 3306, socket = ''): Lost connection
to MySQL server at 'reading initial communication packet', system error: 0 (mysql-errno =
2013)
```

(Bug#42044)

- There was a mismatch between the contents of an SNMP Trap from MySQL Enterprise Monitor and the definition given in the file `MONITOR.MIB`. (Bug#41912)
- If a copy was made of a standard rule, the resulting Wiki markup was incorrect, resulting in the display of user-interface text containing HTML markup. (Bug#41375)
- Allowing the heat chart rules to be set to unscheduled caused the user interface to appear broken. (Bug#41312)
- When a custom rule requiring disk information was created, for example:

  ```
  [Expression]
  %disk_reads%> THRESHOLD

  [Thresholds]
  Critical Alert: 9000
  Warning Alert: 3000
  Info Alert: 1000

  [Variable Assignment]
  variables: %disk_reads%
  Data Item: os:disk:disk_reads
  Instance: /
  ```

  The following error was written to the file `mysql-service-agent.log`:

  ```
  2007-09-05 17:11:00: (critical) disk-get failed for c0d0p1: 2
  ```

  It therefore appeared that the Monitor Agent was not able to obtain the required information. (Bug#30820)

E.6. Changes in MySQL Enterprise Monitor 2.0.2 (14 January 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.1.

Bugs fixed:

- The COMPLETE SETUP button was missing from the **SETUP** page you were taken to after the current subscription expired. (Bug#41685)
- If the `host-id` of the monitored instance and the `host-id` of the current agent did not match, the agent generated the following error message:

  ```
  Please TRUNCATE TABLE mysql.inventory on this mysql-instance and restart the agent.
  ```

  However, it did not suggest using `sql_log_bin = 0`. This is used for all other actions against this table so that they are not replicated to slaves, each of which has their own copy of this table. (Bug#41673)
- The Agent did not start up when the monitored server had many databases and tables, and was under heavy load. This was because the `trigger_schema` query was taking too long on agent start up. (Bug#41555)
- The Monitor Agent failed to install on Solaris 10 x86. The following error was generated:

  ```
  Installing
  0% ______________ 50% ______________ 100%
  ########################################
  Warning: Problem running post-install step.
  Installation may not complete correctly
  Error running /usr/local/mysqlagent/bin/mysql-monitor-agent
  --defaults-file=/usr/local/mysqlagent/etc/mysql-monitor-agent.ini
  --plugins=agent --agent-generate-uuid=true : 2008-12-12 13:06:02: (critical)
  Conversion from character set '646' to 'UTF-8' is not supported
  ```

```
2008-12-12 13:06:02: (message) shutting down normally
```

(Bug#41445)

- The console/stdout appender remained in the log4j configuration, which meant that all the MySQL Enterprise Monitor server logs were duplicated to Catalina's stdout, and thus `catalina.out`, which was wasteful, especially as that file was not rotated or managed. (Bug#41439)

- When creating new multiple user accounts, the first attempt worked fine. However, following attempts to create new users did not show the **Query Analyzer Options** in the **Create User** pop-up until the role field was changed. (Bug#41430)

- When login privileges were required the Service Manager did not redirect the user to the login page. This resulted in error messages being displayed rather than simply redirecting the user to the login page. This problem typically occurred if it was necessary to restart Tomcat. (Bug#41320)

- The monitor 2.0.0.7105 and 2.0.0.7122 Solaris Intel update installer quits unexpectedly. The installer exits from the GUI in the **Backup of Previous Installation** screen, when OpenSolaris is running on top of Sun xVM.

  The console output for both installer versions is given below:

  ```
  shell> ./mysqlmonitor-2.0.0.7105-solaris-intel-update-installer.bin
  X Error of failed request:  BadMatch (invalid parameter attributes)
    Major opcode of failed request:  73 (X_GetImage)
    Serial number of failed request:  21161
    Current serial number in output stream:  21161
  ```

  ```
  shell> ./mysqlmonitor-2.0.0.7122-solaris-intel-update-installer.bin
  X Error of failed request:  BadMatch (invalid parameter attributes)
    Major opcode of failed request:  73 (X_GetImage)
    Serial number of failed request:  21148
    Current serial number in output stream:  21148
  ```

  (Bug#41315)

- The **Rename Group** function failed if the new group name was different to the current one only in the case used; for example, if “Merlin” was changed to “MERLIN”.

  The error generated was:

  ```
  U0105 This group name is already in use. Enter a different name.
  ```

  (Bug#41266)

- The Agent returned an inventory list of all databases and tables. This information was not used by MySQL Enterprise Monitor, other than to populate the `inv_databases` and `inv_tables` tables. For large-scale deployments, where there were many databases and tables, this resulted in redundant XML messages being sent from the Agent to the Service Manager. (Bug#33150)

E.7. Changes in MySQL Enterprise Monitor 2.0.1 (15 December 2008)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.0.

Bugs fixed:

- Having an installation with over 420 master binlogs resulted in the `mysql::status` graph not being plotted. (Bug#41314)

- The MySQL Enterprise Monitor installer generated a segfault on RedHat Fedora 10:

  ```
  shell> ./mysqlmonitor-2.0.0.7105-linux-x86-installer.bin
  Segmentation fault
  ```

  (Bug#41196)

E.8. Changes in MySQL Enterprise Monitor 2.0.0 (11 December 2008)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 1.3.3.

Functionality added or changed:

- **Important Change**: The `server-name` configuration parameter is deprecated. For compatibility, during an upgrade, the information will be migrated to a `displayname` configuration parameter within the individual instance configuration files. This configuration parameter is provided only for compatibility, as display name information is now stored within the main repository. Support for `displayname` is also deprecated and will be removed in a future release.

- **Important Note**: The rules “32-Bit Binary Running on 64-Bit AMD Or Intel System” and “Key Buffer Size Greater Than 4 GB” occasionally do not evaluate correctly due to timing issues. This causes them to be displayed with the **Severity** level of “Unknown”. This is a known issue and will be resolved in future versions of MySQL Enterprise Monitor.

- **Important Note**: When you start the Merlin 2.0 agent from the command line on Windows, you get the following error dialog:

  ```
  "mysql-proxy.exe - Entry Point Not Found"
  "The procedure entry point libiconv_set_relocation_prefix could not be located in the
  dynamic link library iconv.dll"
  ```

  If you click OK the agent works fine after that.

  This only occurs when starting the agent from the command line, and only when there is another version of one of the DLLs that the agent uses somewhere on the current path. This error can be avoided by opening a command prompt, typing `SET PATH=` to clear the path, and then starting the agent.

- The following has been added to the Tomcat `config.properties` properties file:

  ```
       # max connections in the pool for the repository
       default.maxActive=70
  ```

  (Bug#40652)

- The dashboard could be used to change the agent password to one containing the `@` character, or other special characters, which subsequently caused errors. To fix this problem, special characters in passwords are now prevented by the dashboard. The list of special characters not permitted can be found at the following location: http://en.wikipedia.org/wiki/Percent-encoding#Types_of_URI_characters (Bug#37172)

- The **Query Analysis** page was missing the **Refresh** dropdown control that the **Monitor**, **Events**, and **Graphs** pages had at the top. (Bug#36831)

- The User Interface only returned error strings, without any associated error codes. This meant that if the error string was in a language that the user did not understand, it would be very difficult to determine which error actually occurred.

  The User Interface now supports error codes, as well as error strings. This change enables easier testing of multiple locales. (Bug#32131)

- The wording was changed on the fading pop-up subscription alert. The text “account” was changed to “subscription”. (Bug#31492)

- The alert thresholds for the Query Cache Advisor were changed:

	Information	Warning	Critical
From	95	85	75
To	60	50	40

  (Bug#30969)

- The agent log file name has changed from `mysql-service-agent.log` to `mysql-monitor-agent.log`. The old log file will be retained during a upgrade install.

Bugs fixed:

- Starting a new agent against an instance that contained many databases broke up the initial discovery packet, causing collections such as CPU usage and their graphs to fail. See also Bug#33150 and Bug#41314. (Bug#41933)

- When altering an existing rule you had to add an empty string, `""`, to any threshold level that was empty. Otherwise, the rule failed to run and the resulting exceptions caused the **Events** page to be unusable due to duplicate key exceptions. (Bug#41310)

- After installing the 2.0.0.7119 Dashboard the following error was generated in the logs:

  ```
  com.opensymphony.xwork2.util.OgnlValueStack  Warning  Dec 5, 2008 10:41:26 AM  Caught
  an Ognl exception while getting property titleAddition
  ```

  (Bug#41252)

- The `dc_ng_long_now` table became very large partly due to an unused column `begin_time`. (Bug#41093)

- Although there was a unique constraint on the user name, it was not enforced during first-time setup. This resulted in a stack trace being produced, rather than a more user-friendly error message, if the same name was used for the `admin` and `agent` accounts. (Bug#40870)

- MySQL Enterprise Monitor server had stopped sending up/down SNMP traps. (Bug#40861)

- The Query Analyzer's **Explain Query** tab did not have pop up text or a link to the documentation regarding `SELECT` queries. (Bug#40841)

- When you tried to import a Trial-level advisor JAR using a Trial user account, one of the following error messages was generated:

  ```
  U0009 The uploaded Advisor jar was invalid
  ```

  ```
  U0161 Please import a Platinum level Advisor .jar to use with this Platinum level product key
  ```

  (Bug#40834)

- Meta Info on the Dashboard did not display information for the metadata `os_description`. (Bug#40830)

- Attempting to create an alias of statements such as `COMMIT`, `ROLLBACK`, `BEGIN`, resulted in the error:

  ```
  Can't find template commitTnx.st; group hierarchy is [HTMLFormatting]
  org.antlr.stringtemplate.StringTemplateGroup.lookupTemplate(StringTemplateGroup.java:507)
  ```

  (Bug#40736)

- Trying to upgrade from 2.0.0.7088 to 2.0.0.7092 failed as there was a missing file. When the update program `mysqlmonitor-2.0.0.7092-solaris-intel-update-installer.bin` was run, the file `/tmp/com/mysql/merlin/server/version.props` could not be found. (Bug#40692)

- On OS X with Java 1.5, Tomcat crashed on launch with the error:

  ```
  Invalid memory access of location 00000007 eip=013df179
  ```

  (Bug#40689)

- When the agent was installed using the command line installer and `--enableproxy 0` was specified, the installer should have removed `quan.lua` from the `agent-item-files` option in the INI file. (Bug#40551)

- The Agent could not connect to a database with the `hostname` set to `localhost`. Doing so resulted in the error:

  ```
  (critical) the MySQL server could not be reached at socket '(null)', we will check in 10 seconds
  ```

  (Bug#40530)

- The update installer for the 2.0 Monitor did not have an **Is SSL support required?** checkbox. Therefore, the appropriate SSL connector definition was commented out in the `conf/server.xml` file. (Bug#40414)

- The Service Manager installer did not uninstall or wipe out the previous installation if you answered “Yes” to the question:

"The directory you selected already contains a MySQL installation. If you continue installation all data will be lost. Do you wish to continue?" (Bug#40410)

- If you unchecked the **Enable Proxy** checkbox on the **Query Analysis Configuration** screen, the agent's INI file still contained proxy configuration data and was not commented out. (Bug#40272)
- As different queries were sent through the agent it used increasing amounts of memory. (Bug#40260)
- The Service Manager installer set the Java Virtual Machine option `HeapDumpPath` to `<install_root>\temp` on Windows and `/tmp` on other platforms. For consistency the `HeapDumpPath` directory should have been set to `<install_root>\tmp` on Windows. (Bug#40215)
- When using the command line agent setup program, a socket file was not accepted for the monitored instance. (Bug#40085)
- The language option when installing the MySQL Enterprise Monitor in Japanese using the command-line instllaer has been changed from `jp` to `ja`. (Bug#40082)
- When monitoring MySQL 5.1.24 and above, the user used by the agent to connect to the MySQL server for monitoring must have the `PROCESS` privilege. (Bug#40050)
- The check box option string "Is SSL support required?", on the Tomcat Server Option dialog of the Monitor installation, was not correctly translated into Japanese. (Bug#39814)
- The base application directory for the MySQL Enterprise Dashboard has been updated from `http://localhost:18080/merlin` to `http://localhost:18080/`. (Bug#39403)
- If the Service Manager or the MySQL Server running the Repository crashed, they did not restart automatically. (Bug#39377)
- If the agent crashed, there was no watchdog, angel or keep-alive mechanism to restart the agent and keep it running. (Bug#39374)
- If the MySQL client libraries were located in a nonstandard location, the agent 2.0.0.7042 installer failed with a "library not found" error. (Bug#39317)
- For the rule "Server-Enforced Data Integrity Checking Not Strict", the Recommended Action did not display correctly. It displayed as `SET sql_mode=modes`, rather than `SET [GLOBAL|SESSION] sql_mode=modes`. (Bug#39261)
- A query that was issued through the proxy, and that had an auto-explain performed on that query, did not give the correct response to a subsequent query of `SELECT FOUND_ROWS()`. (Bug#39223)
- It was not possible to establish a connection with the Dashboard using the SSL protocol (https://). (Bug#39198)
- When a 1.3 MySQL Enterprise Monitor installation with many rules scheduled was upgraded to 2.0, the upgraded installation was then found to have only the heat chart rules scheduled. (Bug#39043)
- The agent installer did not permit a second agent to be installed on Windows. (Bug#38976)
- The default display name for a host supplied by the agent has been changed. If networking has been disabled in the server, the default display name shown is either `host:socket` for scoket connections, or `host:uuid` for everything else. If networking is enabled, the `host:port` combination is used. If the agent has been configured to use the `displayname` option, this overrides all other default settings. (Bug#38897)
- The Dashboard incorrectly diplayed that insufficient licenses were available, even though sufficient licenses had been purchased. (Bug#38514)
- After running the MySQL Service Agent uninstall program, the file `/etc/init.d/mysql-service-agent` remained present on the server. (Bug#38490)
- The notice fader continued to display English text after you changed the locale to Japanese. (Bug#38460)
- The **Subscription Expired** pop-up window referred to the "Global Preferences" page, instead of the "Global Settings" page. (Bug#38358)
- An inappropriate time zone was used to parse user-entered date and time strings. (Bug#38323)
- A sigar network stats error was generated on the Solaris platform:

```
# /opt/mysql/enterprise/agent/bin/mysql-service-agent --version
MySQL Service Agent - 1.2.0.7879, (glib lib=2.8.5, headers=2.8.5)

SunOS mysqlprd01 5.10 Generic_127127-11 sun4v sparc SUNW,T5240

2008-07-21 10:07:24: (critical) sigar_net_interface_config_primary_get() failed: 6
2008-07-21 10:08:00: (critical) sigar_net_interface_config_primary_get() failed: 6

# /opt/mysql/enterprise/agent/bin/sigar-test-all >/tmp/test.txt
sigar_net_interface_stat_get(e1000g0:2) failed#
```

(Bug#38302)

- After deleting a server from the **Settings**, **Manage Servers** tab, at the very bottom of the page the **Monitoring x instances on x host** values did not reflect the deletion. (Bug#38225)

- The MySQL Enterprise Monitor alert “INFO Alert - Users Can View All Databases On MySQL Server (v 1.5 *)” from the Security advisor was incorrect. This is because the default server behavior permits users to see databases for which they have privileges, not “all databases on server” as suggested by the alert. (Bug#38052)

- The “Maximum Connection Limit Nearing Or Reached” advisor did not generate a new Critical Alert event when there was an open info success event. (Bug#37816)

- The Linux IA64 installer appeared to crash. The installer appeared to crash on RH4_IA64 if called with option "--version":

  ```
  -------------------------------------------------------------------------------
  <INSTALLER> --version
  mysqlmonitorage(30704): unaligned access to 0x6000000000a8413c, ip=0x2000000003ddd5f0
  mysqlmonitorage(30704): unaligned access to 0x6000000000a84144, ip=0x2000000003ddd5f1
  mysqlmonitorage(30704): unaligned access to 0x6000000000a8414c, ip=0x2000000003ddd600
  mysqlmonitorage(30704): unaligned access to 0x6000000000a84154, ip=0x2000000003ddd601
  MySQL Enterprise Monitor Agent 0.7.0.1737 --- Built on 2008-06-25 19:31:53

  -------------------------------------------------------------------------------
  ```

  However, this warning is harmless and will not impact the operation of the agent. (Bug#37496)

- If the log-level option in the agent configuration file was set to an unknown level by mistake, the `init.d` script appeared to enter an infinite loop. (Bug#37108)

- Malformatted server meta information appeared on the Dashboard; **RAM** and **Disk Space** appeared under the **CPU** category. (Bug#36740)

- A “rename” link incorrectly appeared next to the **Upgrade** category on the **Manage Rules** page. (Bug#36584)

- In the case where exceptions were passed through to the User Interface, the substituted arguments in the message contained developer-only information. (Bug#36580)

- **Agent Version**, **Last MySQL Contact**, **OS Info**, **CPU Info**, and **IP Addresses** were all blank on the dashboard when the agent for the selected server was not functioning. (Bug#36301)

- `mysql-monitor-agent` became confused by the `.DS_Store` files that are created on Mac OS X. (Bug#36216)

- The rule “Key Buffer Size Greater Than 4 GB” incorrectly triggered the following alert:

  `CRITICAL Alert - Key Buffer Size Greater Than 4 GB`

  However, on non-Windows systems, a key buffer size greater than 4 GB is supported. (Bug#36143)

- Since the repository database for MySQL Enterprise Monitor uses InnoDB there was no way to reduce the size of the data files after an old log/event data purge operation. Further, the purge data operation executed once per day, and had no option to trigger the purge operation manually. (Bug#35971)

- On the Graphs page, if all graphs were expanded, then the Time Display interval updated, the page was refreshed with the Expand All button displayed, even though all the graphs were already expanded. (Bug#35917, Bug#35133)

- The Meta Info area of the Monitor page incorrectly reported the operating system version number for the MySQL version. (Bug#35836)

- The rule “XA Distributed Transaction Support Enabled For InnoDB” incorrectly sent a warning when the binary log was on. (Bug#35786)
- On the Monitor page, the time displayed for `Last MySQL Contact` lagged behind that for `Last Agent Contact` by a large amount. (Bug#35774)
- MySQL Enterprise Monitor did not update replication settings correctly. After a slave became the master, the Adviser still referred to it as a slave. (Bug#35771)
- The Adviser email suggested using the `--log-queries-not-using-indexes` option. However, this option is not available in MySQL Server versions prior to 4.1. (Bug#35770)
- Thumbnail graphs did not update properly after a time zone change. (Bug#35756)
- If a system had a global `wait_timeout` lower than the general activity of the agent, the agent was disconnected. The monitored server then logged an error and incremented `Aborted_clients`. (Bug#35648)
- Alerts fired after a blackout period based on data collections that happened during the blackout. (Bug#35617)
- The translation of the ADD ROW button on the Rule Definition window was incorrect. (Bug#35495)
- An uninstallation message asked about removing Apache files, even though Apache is no longer used. (Bug#35154)
- After updating from a previous version to the latest 1.3 version, the **QUERY CACHE HAS SUB-OPTIMAL HIT RATE** was still displayed in English after setting the locale to Japanese. Note, the rule was translated correctly if the full installer was used. (Bug#35134)
- The MySQL Enterprise Monitor uninstall dialog box had missing text when using the Japanese locale. (Bug#34982)
- Running the installer with the `--help` option caused an incorrect message to be displayed. (Bug#34200)
- When using the unattended `unInstall` script on Linux together with the option `--env deleteUserData=yes` the correct warning text was displayed. However, this text should not be displayed in unattended mode. Further, the option `--env deleteUserData=yes` was not displayed by the `--help` output. (Bug#34071)
- Platinum Unlimited customers sometimes received a warning stating incorrectly that their subscription supported zero hosts. (Bug#34010)
- Clicking the resolution notes link for a closed event on the events tab showed incorrect behavior. The pop-up initially showed the resolution notes, but when the resolution tab was clicked the notes disappeared and were replaced by an edit box. (Bug#33935)
- The status on the product information page was not translated when the user locale was set to Japanese. (Bug#32785)
- When the locale was set to Japanese, the date picker still had English month titles. (Bug#32741)
- Flashing display of a pop-up used while saving outgoing email settings was caused by problematic initial placement calculations. (Bug#32579)
- AIX 5.2 Agent did not work on AIX 5.3. (Bug#32414)
- When the First Time Setup program was run it did not prompt the user to enable importing an Advisor bundle. (Bug#32199)
- Agent on MacOSX did not read IP addresses for network interfaces correctly, so the monitor displayed empty host IP addresses. (Bug#32188)
- HTML code in queries was not escaped when reporting replication errors, causing the code to be rendered into the page. (Bug#32186)
- The First Run pop-up defaulted to English rather than to the locale set in the browser. (Bug#32129)
- The error dialog box flashed in the upper left corner before being positioned in the center of the screen. This error dialog box now opens in the center of the screen. (Bug#32068)
- The Events list did not take into account Daylight Time and Standard Time when listing events that happened during 1:00am-1:59am. An event that occurred at 1:10am Standard Time was listed before an event that occurred 50 minutes before it at 1:20am Daylight Time. (Bug#32016)

- The pop-up for editing log levels failed to load due to bad instantiation data. (Bug#32013)
- During the repeated hour of Daylight Savings Time (when 2am turns back into 1am), the graphs were not drawing data. Instead, there was a straight line from the point at 1:00 to the second 1:00, which is what happens if there is no data. The repository did, however, have data for this hour. (Bug#31997)
- Only US English was supported for a locale setting. Other English variants are now available for the `locale` setting on the `General Settings` or the `User Preferences` pages. (Bug#31801)
- If the user locale was changed the graph cache would continue to display the graph in the last locale until it timed out. (Bug#31680)
- No init script was installed for the MySQL Network Monitoring Service Manager, and so it did not restart automatically on reboot. (Bug#31676)
- The graph's displayed time was not the local time of the Dashboard corresponding to the requested time on the monitored server. (Bug#31656)
- Saving a rule with a name that already existed resulted in a stack trace in the window, instead of a more user-friendly error message. (Bug#30925)
- The `network.mysql.com` error messages were remapped thereby causing confusion. For example, the following error message:

  E9000: MySQL Enterprise Customer Center is having difficulties fetching your contract information. Please contact enterprise-feedback@mysql.com for assistance.

  Was remapped to:

  Unable to connect to verify credentials (Bug#30873)
- A newly added server showed as “down” in the user interface, and could potentially have sent a false alarm notification. (Bug#30735)
- The information on the Advisors, Check For Updates page did not accurately reflect how many rules and graphs were actually in the database and available to the user. (Bug#29623)
- The agent did not process `SIGHUP`. (Bug#29380)
- Monitor did not have a facility to stop or downgrade an agent collection frequency. (Bug#28589)
- After an agent installation was updated from 1.0.1.4391 to 1.1.0.4899, the version in the Add/Remove Programs menu was incorrectly displayed as 1.0.1.4391, even though the update was successful and the file version of `agent.exe` was correctly displayed as 1.1.0.4899. (Bug#27447)
- When viewing the **Results** of an **Event** in the **Events** tab of the **Dashboard**, the **Notifications** section did not reflect the **Notifications** settings at the time the **Event** was triggered, but rather the **Notifications** settings at the time the **Event Results** were viewed. (Bug#26349)
- When the Monitor Agent was remotely monitoring a MySQL server it incorrectly reported that it could collect operating system information. (Bug#22497)
- The Account Without Password advisor did not report all users who were without a password, it only reported one. (Bug#15165)

Appendix F. MySQL Enterprise Monitor Reference

This appendix contains reference information for MySQL Enterprise Monitor.

F.1. MySQL Enterprise Monitor Limitations

The following list provides information on known limitations within MySQL Enterprise Monitor

- The maximum size for data stored within the `mysql.inventory` table is 64 bytes.
- The MySQL Enterprise Monitor Agent when operating for Query Analyzer as a proxy cannot handle queries greater than 16MB.
- The MySQL Enterprise Monitor Agent when operating for Query Analyzer as a proxy does not support clients using the MySQL protocol older than MySQL 5.0.
- The MySQL Enterprise Monitor Agent when operating for Query Analyzer as a proxy does not support clients affects the authentication of clients. As the proxy is the real client connecting to the backend MySQL server, authentication should use the hostname of the proxy, not the original client.
- When viewing the graphs generated by MySQL Enterprise Service Manager, the fonts used for the graphs may not be displayed properly if the required font is missing.

  You can specify a custom font by setting a custom property within the embedded MySQL Server to a known working font. To do this, connect to your embedded MySQL sever using the information provided by the `configuration_report.txt` file. Once connected, execute the following statment:

  ```
  mysql> INSERT INTO map_entries VALUES (1,'Helvetica','graph.font');
  ```

  Replacing `Helvetica` with the font you want to use.

  The font used for graph rendering will be selected as follows, in this order:

  1. The user override value.
  2. The MySQL Enterprise Service Manager default of Arial.
  3. The graph engine default of SansSerif.
- It is not possible to monitor more than one MySQL Enterprise Dashboard instance hosted on the same machine simultaneously within the same browser, even if each MySQL Enterprise Dashboard uses a different port for access. The effect of this limitation may be more significant on certain browsers.
- When monitoring multiple MySQL instances with one MySQL Enterprise Monitor Agent, if any of the MySQL instances are down at the point when the agent is started, the agent will not attempt to reconnect to the servers. This could lead to indications that MySQL instances were down when in fact they were running and available. To avoid this problem, ensure all of the monitored instances are available before starting the agent.

F.2. Supported Browsers

The MySQL Enterprise Dashboard is known to work within the following browser environments:

- Microsoft Internet Explorer 7.x, 8.x

  > **Note**
  >
  > MySQL Enterprise Dashboard is not compatible with the **Compatibility View** option within Internet Explorer 8.x. Please make sure this option is disabled before using MySQL Enterprise Dashboard
- Safari 3.2, 4.x

- Firefox 3.0, 3.5

The MySQL Enterprise Dashboard is known *not* to work within the following browser environments:

- Microsoft Internet Explorer 6.x
- Opera
- Google Chrome

F.3. Installation Requirements

The Service Manager is available for Windows, Mac OS X, and a variety of Unix and Linux operating systems. In most cases, the standard operating system and packages are supported without further libraries and modules. All the required components are installed automatically.

- **Mac OS X Notes**
  - The Mac OS X Service Manager is only supported on Intel architecture. However, the Mac OS X agent is supported for both Intel and the PowerPC.
  - Installation requires approximately 450MB of disk space for MySQL Enterprise Service Manager
- **Windows Notes**
  - Installation requires approximately 260MB of disk space for MySQL Enterprise Service Manager
- **Unix/Linux Notes**
  - Installation requires approximately 450MB of disk space for MySQL Enterprise Service Manager
  - On FreeBSD, you must have `bind8` installed.

> ### Note
>
> Disk-space usage requirements are approximate for the base application. During usage, the service manager records detailed information about your MySQL servers and environment, and disk space usage will increase with time to record this historical data. You can control how long information is stored once the service manager has been installed.

The minimum recommended requirements for the service manager are at least a 2GHz CPU, with two or more CPU cores, and at least 2GB of RAM. If you are monitoring a large number of services, then there will be an increased load on the server manager. Running the service manager on a machine that is already running other tasks is only recommended if you are monitoring a small number of agents. For monitoring five or more agents simultaneously, you should dedicate a machine to the process. For more information, see Section F.5, “Choosing Suitable MySQL Enterprise Service Manager Hardware Configurations”.

The Monitor Agent is available for a wide range of operating systems. For an up-to-date list please see the http://www.mysql.com/products/enterprise/. The agent can be used to monitor any MySQL server from version 4.0.x through 6.0.x.

Production support is not currently available for MySQL Enterprise Monitor (Service Manager component) in virtualized environments.

F.4. Creating a new SSL KeyStore

The SSL certificate that is provided with the server during installation is a sample certificate only. If you want to use SSL for communicating with your MySQL Enterprise Service Manager and MySQL Enterprise Dashboard you should replace the supplied certificate with your own certificate.

The certificate keystore is located in the `apache-tomcat/conf/myKeyStore` file within your MySQL Enterprise Service Manager installation.

To create a new, self-signed, certificate, use the `keytool` command (provided with Java SE Security) to create a new certificate file:

```
shell> keytool -genkey -alias tomcat -keyalg RSA -validity 1825 -keystore newKeystore
```

You will be prompted to fill in the certificate information and to provide a unique password for the certificate.

The `-validity` specifies the length of the certificate created, specified in days. The `-alias` is required to ensure that the certificate is associated with `tomcat`.

Once you have created the new keystore, you should shutdown MySQL Enterprise Service Manager, copy the new keystore file to `apache-tomcat/conf/myKeyStore` and restart MySQL Enterprise Service Manager.

If you have an existing certificate that you would like to import into your keystore, you need to import the certificate using the following command:

```
shell> keytool -import -alias tomcat -keystore newKeystore -file public.pem
```

For more information on using SSL certificates in Tomcat see Apache Tomcat 6.0: SSL Configuration HOW-TO.

F.5. Choosing Suitable MySQL Enterprise Service Manager Hardware Configurations

Running MySQL Enterprise Service Manager places a reasonable load on your system, and this load increases linearly as you add more agents monitoring more servers. Ideally, you should use a dedicated machine for MySQL Enterprise Service Manager, rather than running it alongside other applications and services.

Minimum System Requirements

- 2 or more CPU cores
- 2 or more GB of RAM
- Disk I/O subsystem applicable for a write-intensive database

Recommended System Requirements (if monitoring 100 or more MySQL servers)

- 4 or more CPU cores
- 8 or more GB of RAM
- Disk I/O subsystem applicable for a write-intensive database (RAID10, RAID 0+1)

F.6. MySQL Enterprise Monitor Agent Reference

The MySQL Enterprise Monitor Agent supports the following configurable options:

Table F.1. `mysql-monitor-agent` Help Options

Format	Config File	Description	Introduction	Deprecated	Removed
--help	help	Show help options	2.0.0		
--help-admin	help-admin	Show options for the admin-module	2.0.0		
--help-agent	help-agent	Show agent options	2.0.0		
--help-all	help-all	Show all help options	2.0.0		
--help-monitor	help-monitor	Show monitor options	2.0.0		
--help-proxy	help-proxy	Show options for the proxy-module	2.0.0		

Table F.2. `mysql-monitor-agent` Admin Options

Format	Config File	Description	Introduction	Deprecated	Removed
--admin-address	admin-address	Defines the listening address and poirt for the admin module	2.0.0		
--admin-lua-script	admin-lua-script	Script to execute by the admin module	2.0.0		
--admin-password	admin-password	Password for authentication for admin module	2.0.0		
--admin-username	admin-username	Username for authentication for admin module	2.0.0		
--proxy-address	proxy-address	Listening address:port of the proxy server	2.0.0		

Table F.3. `mysql-monitor-agent` Agent Options

Format	Config File	Description	Introduction	Deprecated	Removed
--agent-collector-plugins	agent-collector-plugins	Load the specified collector plugins	2.0.0		
--agent-generate-uuid	agent-generate-uuid	Generate a UUID for use with the agent-uuid	2.0.0		
--agent-heartbeat-interval	agent-heartbeat-interval	The interval for generating heartbeat operations	2.0.0		
--agent-host-id	agent-host-id	The host ID for the agent	2.0.0		
--agent-host-id-commandline	agent-host-id-commandline	The command to use to generate the agent host id	2.0.0		
--agent-hostname	agent-hostname	Hostname of the agent host	2.0.0		
--agent-item-files	agent-item-files	List of data items for additional collections	2.0.0		
--agent-mgmt-hostname	agent-mgmt-hostname	URL of the management server	2.0.0		
--agent-run-os-tests	agent-run-os-tests	Run the operating system tests and shutdown	2.0.0		
--agent-sync-attempts	agent-sync-attempts	Synchronize the attempts to connect at resync	2.0.0		
--agent-track-alloc	agent-track-alloc	Enable the tracking of the allocation sizes	2.0.0		
--agent-uuid	agent-uuid	UUID of this agent	2.0.0		

Table F.4. `mysql-monitor-agent` Monitor Options

Format	Config File	Description	Introduction	Deprecated	Removed
--monitor-interval	monitor-interval	Interval for executing queries against the backend	2.0.0		

Format	Config File	Description	Introduction	Deprecated	Removed
--monitor-lua-script	monitor-lua-script	Script filename for the monitor	2.0.0		
--monitor-password	monitor-password	Set the password for the monitored MySQL Server	2.0.0		
--monitor-username	monitor-username	Set the username for the monitored MySQL Server	2.0.0		

Table F.5. `mysql-monitor-agent` Proxy Options

Format	Config File	Description	Introduction	Deprecated	Removed
--no-proxy	no-proxy	Don't start the proxy module	2.0.0		
--proxy-backend-addresses	proxy-backend-addresses	Address:port of the remote MySQL server	2.0.0		
--proxy-fix-bug-25371	proxy-fix-bug-25371	Enable the fix for Bug#25371 (for mysqld > 2.0.0.12) for older libmysql versions	2.0.0		
--proxy-lua-script	proxy-lua-script	Filename for Lua script for proxy operations	2.0.0		
--proxy-pool-no-change-user	proxy-pool-no-change-user	Don't use the protocol CHANGE_USER to reset the connection when coming from the connection pool	2.0.0		
--proxy-read-only-backend-addresses	proxy-read-only-backend-addresses	Address:port of the remote MySQL server (read only)	2.0.0		
--proxy-skip-profiling	proxy-skip-profiling	Disabled profiling of queries	2.0.0		

Table F.6. `mysql-monitor-agent` Application Options

Format	Config File	Description	Introduction	Deprecated	Removed
--basedir	basedir	Specify the base directory to prepend to paths in the config	2.0.0		
--daemon	daemon	Start in daemon-mode	2.0.0		
--defaults-file	defaults-file	Specify the configuration file	2.0.0		
--keepalive	keepalive	Try to restart the proxy if a crash occurs	2.0.0		
--log-backtrace-on-crash	log-backtrace-on-crash	Try to invoke the debug and generate a backtrace on crash	2.0.0		
--log-file	log-file	Specify the file for logging error messages	2.0.0		
--log-level	log-level	Logging level	2.0.0		

Format	Config File	Description	Introduction	Deprecated	Removed
--log-use-syslog	log-use-syslog	Log errors to syslog	2.0.0		
--lua-cpath	lua-cpath	Set the LUA_CPATH	2.0.0		
--lua-path	lua-path	Set the LUA_PATH	2.0.0		
--max-open-files	max-open-files	Specify the maximum number of open files to support	2.0.0		
--pid-file	pid-file	PID file to store the process ID (when in daemon mode)	2.0.0		
--plugin-dir	plugin-dir	Path to the plugin files	2.0.0		
--plugins	plugins	List of plugins to load	2.0.0		
--user	user	Specify the user to use when running mysql-monitor-agent	2.0.0		
--version	version	Show the version information	2.0.0		

- `--help`

Version Introduced	2.0.0
Command-Line Format	`--help`
	`-?`

  Show available help options.

- `--help-all`

Version Introduced	2.0.0
Command-Line Format	`--help-all`

  Show all help options.

- `--help-admin`

Version Introduced	2.0.0
Command-Line Format	`--help-admin`

  Show options for the admin-module.

- `--help-proxy`

Version Introduced	2.0.0
Command-Line Format	`--help-proxy`

  Show options for the proxy-module.

- `--help-monitor`

Version Introduced	2.0.0
Command-Line Format	`--help-monitor`

  Show options for the monitor module.

- `--help-agent`

Version Introduced	2.0.0	
Command-Line Format	`--help-agent`	

Show options for the agent module.

- `--admin-address=host:port`

Version Introduced	2.0.0	
Command-Line Format	`--admin-address`	
Config-File Format	`admin-address`	
	Permitted Values	
	Type	`string`
	Default	`:4041`

Specify the host name (or IP address) and port for the administration port. The default is `localhost:4041`.

- `--admin-lua-script=script`

Version Introduced	2.0.0	
Command-Line Format	`--admin-lua-script`	
Config-File Format	`admin-lua-script`	
	Permitted Values	
	Type	`filename`
	Default	

Specify the script to use for the administration module for the proxy.

- `--admin-password=pass`

Version Introduced	2.0.0	
Command-Line Format	`--admin-password`	
Config-File Format	`admin-password`	
	Permitted Values	
	Type	`string`
	Default	

Define the password to use to authenticate users wanting to connect to the administration module of `mysql-monitor-agent`. The administration module uses the MySQL protocol to request a username and password for connections.

- `--admin-username=user`

Version Introduced	2.0.0	
Command-Line Format	`--admin-username`	
Config-File Format	`admin-username`	
	Permitted Values	
	Type	`string`
	Default	`root`

Define the username to use to authenticate users wanting to connect to the administration module of `mysql-monitor-agent`. The default username is `root`. The administration module uses the MySQL protocol to request a username and password for connections.

- `--agent-collector-plugins=user`

Version Introduced	2.0.0	
Command-Line Format	`--agent-collector-plugins`	
Config-File Format	`agent-collector-plugins`	
	Permitted Values	
	Type	`numeric`
	Default	`collect_lua`

A comma-separated list of the plugins to be used and loaded by the agent and used to provide collector items.

- `--agent-generate-uuid`

Version Introduced	2.0.0
Command-Line Format	`--agent-generate-uuid`
Config-File Format	`agent-generate-uuid`

Regenerates a UUID for the agent. You can use this to generate a new UUID if you have copied configuration information for an agent from one machine to another. To use:

```
shell> /opt/mysql/enterprise/agent/bin/mysql-monitor-agent --agent-generate-uuid
ee9296d7-f7cd-4fee-8b26-ead884ebf398
2009-03-05 11:49:37: (critical) shutting down normally
```

For more information, see Section 2.3.6.3, “Monitoring Multiple MySQL Servers”.

- `--agent-heartbeat-interval=#`

Version Introduced	2.0.0	
Command-Line Format	`--agent-heartbeat-interval`	
Config-File Format	`agent-heartbeat-interval`	
	Permitted Values	
	Type	`numeric`
	Default	`3`

Specify the heartbeat interval. The agent sends heartbeat messages to MySQL Enterprise Service Manager to indicate that it is available and still running and monitoring servers. The MySQL Enterprise Service Manager uses this information to help identify when an agent is available or not.

- `--agent-host-id=HOSTID`

Version Introduced	2.0.0
Command-Line Format	`--agent-host-id`
Config-File Format	`agent-host-id`

Define the agent host id. The host id should be a unique identifier for the host on which the agent is running, and is different to the agent UUID. You can override the host ID for use within HA environments where you have two identical agent configurations, one on a live server, and one on a backup server to be brought into use if the primary server fails. The host ID is used to identify the host by the server, and during failover you want the host ID to remain constant as you monitor either the original or failover (replacement) server.

- `--host-id-commandline`

Version Introduced	2.0.0
Command-Line Format	`--agent-host-id-commandline`
Config-File Format	`agent-host-id-commandline`

Specify the command line to generate the host ID. By default, the host ID is generated by using a variety of methods configured for a specific platform, including using SSH keys and embedded host ID information. You can replace this with an alternative command-line

- `--basedir`

Version Introduced	2.0.0
Command-Line Format	`--agent-hostname`
Config-File Format	`agent-hostname`

Override the automatically determined hostname. You can use this to specify an alternate hostname for the agent. This can be useful if your host has multiple hostnames and you want the machine identified by a specific hostname.

- `--agent-item-files`

Version Introduced	2.0.0
Command-Line Format	`--agent-item-files`
Config-File Format	`agent-item-files`

A comma separated list of the item files used to load data items that are then collected and sent to MySQL Enterprise Service Manager. For more information, see Section 5.4.8, “Creating a Custom Data Collection Item”.

- `--agent-mgmt-hostname`

Version Introduced	2.0.0
Command-Line Format	`--agent-mgmt-hostname`
Config-File Format	`agent-mgmt-hostname`

Set the URL of the MySQL Enterprise Service Manager where collected data is sent. The URL should include the username and password for the agents, for example: `http://agent:password@memserver:18080/heartbeat`.

> **Note**
>
> If you have the `http_proxy` environment variable configured within your environment, you should add the value of `agent-mgmt-hostname` to the `no_proxy` variable to ensure that data is correctly transmitted to MySQL Enterprise Service Manager and not redirected through the configured proxy address.

- `--agent-run-os-tests`

Version Introduced	2.0.0
Command-Line Format	`--agent-run-os-tests`
Config-File Format	`agent-run-os-tests`

Runs the internal operating system tests, and then exits. Using this option will generate a large body of information about the various operating system information that can be collected about the host. You can use this to determine whether the agent is collecting the right amount, quantity and quality of information to the server.

A truncated sample of the output is provided below:

```
2010-01-22 16:15:45: (critical) MySQL Monitor Agent 2.1.1.1138 started.
sigar-test-all.c.128 ():
  pid = 1353
sigar-test-all.c.110 ():
  mem.ram = 6080,
  mem.total = 6374154240,
  mem.used = 3356090368,
  mem.free = 3018063872,
  mem.actual_free = 3018063872,
  mem.actual_used = 3356090368
sigar-test-all.c.143 ():
  swap.total = 28523896832,
  swap.used = 710623232,
  swap.free = 27813273600
sigar-test-all.c.169 ():
  cpu.user = 24544920,
  cpu.sys = 136764840,
  cpu.nice = 0,
  cpu.idle = 1234759920,
  cpu.wait = 0,
  cpu.total = 349015500
sigar-test-all.c.194 ():
[0]
  cpu.user = 8516770,
  cpu.sys = 56838970,
  cpu.nice = 0,
  cpu.idle = 283667220,
  cpu.wait = 0,
  cpu.total = 349022960
[1]
  cpu.user = 6130420,
  cpu.sys = 12671090,
  cpu.nice = 0,
...
```

- `--agent-sync-attempts`

Version Introduced	2.0.0
Command-Line Format	`--agent-sync-attempts`
Config-File Format	`agent-sync-attempts`

Attempt to synchronise with the server during the resynchronization.

- `--basedir`

Version Introduced	2.0.0
Command-Line Format	`--agent-track-alloc`
Config-File Format	`agent-track-alloc`

Track the memory allocation in the various modules of the agent to help monitor the memory usage.

- `--basedir`

Version Introduced	2.0.0	
Command-Line Format	`--agent-uuid`	
Config-File Format	`agent-uuid`	
	Permitted Values	
	Type	`string`

Specify the agent UUID. A UUID is automatically generated for each agent during installation, with the UUID automatically added to the configuration. You can generate a new UUID using the `agent-generate-uuid` command line option.

- `--basedir`

Version Introduced	2.0.0	
Command-Line Format	`--basedir`	
Config-File Format	`basedir`	
	Permitted Values	
	Type	`filename`

Specify a base directory that will be prepended to all other filename configuration options. The base name should be an absolute (not relative) directory. If you specify a relative directory, `mysql-monitor-agent` will generate an error during startup.

- `--defaults-file`

Version Introduced	2.0.0
Command-Line Format	`--defaults-file`
Config-File Format	`defaults-file`

Specify a file to use as the file with configuration information. If not specified, configuration options are only taken from the command line.

- `--event-threads=#`

Command-Line Format	`--event-threads`	
Config-File Format	`event-threads`	
	Permitted Values	
	Type	`numeric`
	Default	`1`

Specify the number of event threads reserved to handle incoming requests.

- `--keepalive`

Version Introduced	2.0.0
Command-Line Format	`--keepalive`
Config-File Format	`keepalive`

Creates a process surrounding the main `mysql-monitor-agent` which will attempt to restart the true `mysql-monitor-agent` process in thr event a crash or other failure.

- `--log-backtrace-on-crash`

Version Introduced	2.0.0
Command-Line Format	`--log-backtrace-on-crash`
Config-File Format	`log-backtrace-on-crash`

Logs the backtrace to the error log and tries to initialize the debugger in the event of a failure.

- `--log-file=filename`

Version Introduced	2.0.0
Command-Line Format	`--log-file`

Config-File Format	`log-file`	
	Permitted Values	
	Type	`filename`

Specify the name of a file to be used to record log information.

- `--log-file=filename`

Version Introduced	2.0.0	
Command-Line Format	`--log-level`	
Config-File Format	`log-level`	
	Permitted Values	
	Type	`enumeration`
	Valid Values	`error`, `warning`, `info`, `message`, `debug`

The log level to be used when outputting error messages. The specification will output that level (or lower) of a given error message. For example, `message` will also output `info`, `warning`, and `error` messages.

- `--log-use-syslog`

Version Introduced	2.0.0
Command-Line Format	`--log-use-syslog`
Config-File Format	`log-use-syslog`

Causes errors to be sent to the syslog (Unix/Linux only).

- `--lua-cpath=path`

Version Introduced	2.0.0	
Command-Line Format	`--lua-cpath`	
Config-File Format	`lua-cpath`	
	Permitted Values	
	Type	`filename`

Specify the `LUA_CPATH` to be used when loading compiled modules or libraries for Lua scripts.

- `--lua-path=path`

Version Introduced	2.0.0	
Command-Line Format	`--lua-path`	
Config-File Format	`lua-path`	
	Permitted Values	
	Type	`filename`

Specify the `LUA_CPATH` to be used when loading modules for Lua.

- `--max-open-files=#`

Version Introduced	2.0.0

Command-Line Format	`--max-open-files`	
Config-File Format	`max-open-files`	
	Permitted Values	
	Type	`numeric`

The maximum number of open files and sockets supported by the `mysql-monitor-agent` process. You may need to increase this with certain scripts.

- `--monitor-interval=#`

Version Introduced	2.0.0	
Command-Line Format	`--monitor-interval`	
Config-File Format	`monitor-interval`	
	Permitted Values	
	Type	`numeric`
	Default	`1`
	Min Value	`1`

Execute queries against the backends at this interval. The default is 1.

- `--monitor-lua-script=SCRIPT`

Version Introduced	2.0.0	
Command-Line Format	`--monitor-lua-script`	
Config-File Format	`monitor-lua-script`	
	Permitted Values	
	Type	`string`

Filename of the Lua script to use for the monitor module.

- `--monitor-password=PASS`

Version Introduced	2.0.0	
Command-Line Format	`--monitor-password`	
Config-File Format	`monitor-password`	
	Permitted Values	
	Type	`string`

The password for the monitoring user account.

- `--monitor-username=USER`

Version Introduced	2.0.0	
Command-Line Format	`--monitor-username`	
Config-File Format	`monitor-username`	
	Permitted Values	
	Type	`string`
	Default	`monitor`

The username for the monitoring user account.

- `--no-proxy`

Version Introduced	2.0.0
Command-Line Format	`--no-proxy`
Config-File Format	`no-proxy`

Disable the proxy module.

- `--plugin-dir=path`

Version Introduced	2.0.0	
Command-Line Format	`--plugin-dir`	
Config-File Format	`plugin-dir`	
	Permitted Values	
	Type	`filename`

The directory to use when loading plugins for `mysql-monitor-agent`.

- `--plugins=plugin,...`

Version Introduced	2.0.0	
Command-Line Format	`--plugins`	
Config-File Format	`plugins`	
	Permitted Values	
	Type	`string`

A comma-separated list of the plugins to be loaded.

- `--proxy-address=host:port`

Version Introduced	2.0.0	
Command-Line Format	`--proxy-address`	
	`-P`	
Config-File Format	`proxy-address`	
	Permitted Values (>= 2.0.0)	
	Type	`string`
	Default	`:4040`

The listening host name (or IP address) and port of the proxy server. The default is `:4040` (all IPs on port 4040).

- `--proxy-read-only-backend-addresses=host:port`

Version Introduced	2.0.0	
Command-Line Format	`--proxy-read-only-backend-addresses`	
	`-r`	
Config-File Format	`proxy-read-only-backend-addresses`	
	Permitted Values	
	Type	`string`

The listening host name (or IP address) and port of the proxy server for read-only connections. The default is for this information not to be set.

- `--proxy-backend-addresses=host:port`

  The host name (or IP address) and port of the MySQL server to connect to. You can specify multiple backend servers by supplying multiple options. Clients are connected to each backend server in round-robin fashion. For example, if you specify two servers A and B, the first client connection will go to server A; the second client connection to server B and the third client connection to server A.

- `--proxy-pool-no-change-user`

Version Introduced	2.0.0
Command-Line Format	`--proxy-pool-no-change-user`
Config-File Format	`proxy-pool-no-change-user`

  Disables the use of the MySQL protocol `CHANGE_USER` when reusing a connection from the pool of connections specified by the `backend-addresses` list.

- `--proxy-skip-profiling`

  disables profiling of queries (tracking time statistics). The default is for tracking to be enabled.

- `--proxy-fix-bug-25371`

Version Introduced	2.0.0
Command-Line Format	`--proxy-fix-bug-25371`
Config-File Format	`proxy-fix-bug-25371`

  Gets round an issue when connecting to a MySQL server later than 5.1.12 when using a MySQL client library of any earlier version.

- `--proxy-lua-script=file`

Version Introduced	2.0.0	
Command-Line Format	`--proxy-lua-script`	
	`-s`	
Config-File Format	`proxy-lua-script`	
	Permitted Values	
	Type	`filename`

  specify the Lua script file to be loaded. Note that the script file is not physically loaded and parsed until a connection is made. Also note that the specified Lua script is reloaded for each connection; if the content of the Lua script changes while `mysql-monitor-agent` is running then the updated content will automatically be used when a new connection is made.

- `--daemon`

Version Introduced	2.0.0
Command-Line Format	`--daemon`
Config-File Format	`daemon`

  Starts the proxy in daemon mode.

- `--pid-file=file`

Version Introduced	2.0.0	
Command-Line Format	`--pid-file`	
Config-File Format	`pid-file`	
	Permitted Values	
	Type	`filename`

Sets the name of the file to be used to store the process ID.

- `--user=user`

Version Introduced	2.0.0	
Command-Line Format	`--user`	
Config-File Format	`user`	
	Permitted Values	
	Type	`string`

Run `mysql-monitor-agent` as the specified `user`.

- `--version`

Version Introduced	2.0.0
Command-Line Format	`--version`
	`-V`
Config-File Format	`version`

Show the version number.

F.7. Configuring Tomcat Parameters

The parameters for the Tomcat hosting system used to support MySQL Enterprise Service Manager and MySQL Enterprise Dashboard can have an affect on the performance of the system.

The default settings for Java for Tomcat are as follows:

Initial heap size	-Xms	768MB
Maximum heap size	-Xmx	768MB
Java stack size	-Xss	128MB

You can change these parameters to higher values,by editing the `JAVA_OPTS` setting within the corresponding platform script.

- On Unix/Linux and Mac OS X

  Edit the values within `apache-tomcat/bin/setenv.sh` file within your MySQL Enterprise Service Manager directory. You should avoid setting maximum figures higher than the physical memory of your system as this may reduce, rather than increase the performace.

  If you change these parameters, you will need to shutdown and restart MySQL Enterprise Service Manager for the changes to take effect.

- Windows

Edit the JAVA_OPTS settings within the apache-tomcat/bin/setenv.bat file.

To enable the changes, you must re-install your service. To do this, shutdown your MySQL Enterprise Service Manager service and then reinstall the service by running:

```
shell> mysqlmonitorctl.bat uninstall
shell> mysqlmonitorctl.bat install
```

On all platforms, changes to the JAVA_OPTS using the above methods should be retained over an upgrade of the MySQL Enterprise Service Manager service.

If you change these parameters, you must restart MySQL Enterprise Service Manager for the changes to take effect.

F.8. Backing up MySQL Enterprise Service Manager

If you want to backup the data stored within your MySQL Enterprise Service Manager, you can use any of the typical backup solutions, such as mysqldump, to save your data. All you need to backup the information is host name, user name and password details that were set during the installation of the MySQL Enterprise Service Manager

You can locate this information by examining the contents of the configuration_report.txt file that was generated when MySQL Enterprise Service Manager was installed. A scample of the file is provided below:

```
MySQL Enterprise Monitor (Version 2.0.0.7088 : 20081031_152749_r7088)

Here are the settings you specified:
Application hostname and port: http://127.0.0.1:18080
Tomcat Ports: 18080 - 18443 (SSL)
MySQL Port : 13306

Repository Credentials (bundled MySQL):
---------------------------------------
service_manager/Password

Use the following command to login to the MySQL Enterprise Monitor database:
mysql -uservice_manager -pPassword -P13306 -h127.0.0.1
```

The last line provides the information about how to connect to the server using the standard mysql command line client.

All the MySQL Enterprise Monitor repository information, including your configuration, rule and historical data is stored within the mem database.

To backup this information using mysqldump you might use the following command:

```
shell> mysqldump --single-transaction »
-uservice_manager -pPassword -P13306 -h127.0.0.1 mem >mem.dump
```

The above command would create a file, mem.dump, containing all of the MySQL Enterprise Monitor data.

To ensure consistency in a recovery situation, you may also want to backup the agent configuration and metadata stored on each monitored MySQL server. To do this:

- Backup the configuration files of each agent. You should keep a copy of the etc directory for each agent. This directory contains the main configuration file, mysql-monitor-agent.ini, and the configuration information for each server being monitored, which is stored within the etc/instances directory.
- On each server being monitored, retain a copy of the mysql.inventory table, which contains the unique ID of the MySQL server.

F.9. Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0

You can migrate the data generated during a MySQL Enterprise Monitor 1.3.x installation using the Data Migration functionality of the **Server Configuration** panel.

To use the data migration feature, you must have installed MySQL Enterprise Service Manager using an *update* installer. The update installer performs the initial migration of your configuration, rules, schedule, and events data. The historical data is not migrated until you explicitly request the migration of information within the **Manage Servers** section of the **Settings** panel.

Data migration works on a single server, allowing you to select on which servers you want to migrate information. The migration is subject to the following:

- You must elect to migrate the data from each server individually.
- Migration takes approximately 5-6 hours, for each month, for each server. Therefore, if you have six months of data on 10 servers it could take between 300 and 360 hours (15 days) to migrate all of your historical data one server at a time.
- To limit the data migration, set the **Data Purge Behavior** within the **Settings** page. Only data more recent than the specified purge period will be migrated. Data older than the purge period will be ignored.
- To prevent performance issues, migrate only one or a small number of servers concurrently.
- You can start and stop the migration of the data at any time. As a general guide, you should avoid stopping the data migration process and allow it to complete unless:
  - Run out of disk space.
  - MySQL Enterprise Service Manager becomes too slow and unresponsive.
  - Migration never completes.

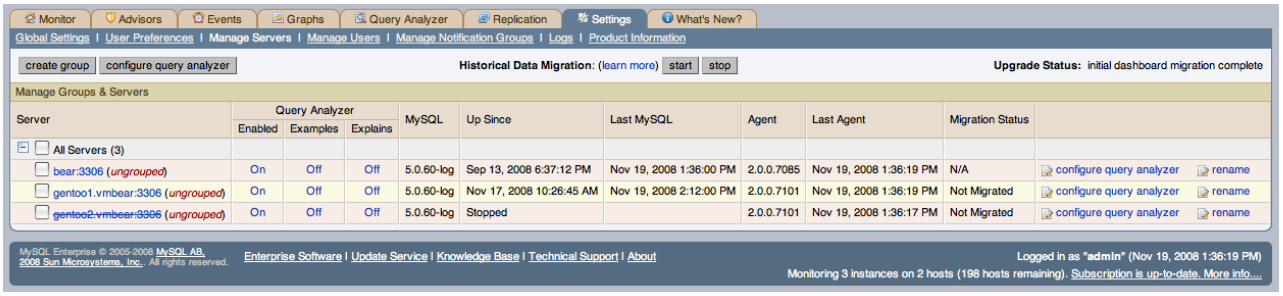
  With the last item, where the migration never completes, occasionally there are some aspects of the data that cannot be migrated successfully. This will prevent the migration process completing, but does not affect the conversion of any data that could be migrated.

Starting Historical Data Migration

To start data migration:

- Switch to the **Manage Servers** display of the **Settings** panel within MySQL Enterprise Dashboard.
- Ensure that the data migration functionality has been enabled. The Start and Stop buttons next to **Historical Data Migration** will be visible.

Figure F.1. MySQL Enterprise Monitor: Historical Data Migration Availability


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New?
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information
create group
configure query analyzer
start
stop

Historical Data Migration: (learn more)

Upgrade Status: initial dashboard migration complete

Manage Groups & Servers

Server	Query Analyzer			MySQL	Up Since	Last MySQL	Agent	Last Agent	Migration Status	
	Enabled	Examples	Explains							
⊟ □ All Servers (3)										
□ bear:3306 (*ungrouped*)	On	Off	Off	5.0.60-log	Sep 13, 2008 6:37:12 PM	Nov 19, 2008 1:36:00 PM	2.0.0.7085	Nov 19, 2008 1:36:19 PM	N/A	configure query analyzer rename
□ gentoo1.vmbear:3306 (*ungrouped*)	On	Off	Off	5.0.60-log	Nov 17, 2008 10:26:45 AM	Nov 19, 2008 2:12:00 PM	2.0.0.7101	Nov 19, 2008 1:36:19 PM	Not Migrated	configure query analyzer rename
□ ~~gentoo2.vmbear:3306~~ (*ungrouped*)	On	Off	Off	5.0.60-log	Stopped		2.0.0.7101	Nov 19, 2008 1:36:17 PM	Not Migrated	configure query analyzer rename

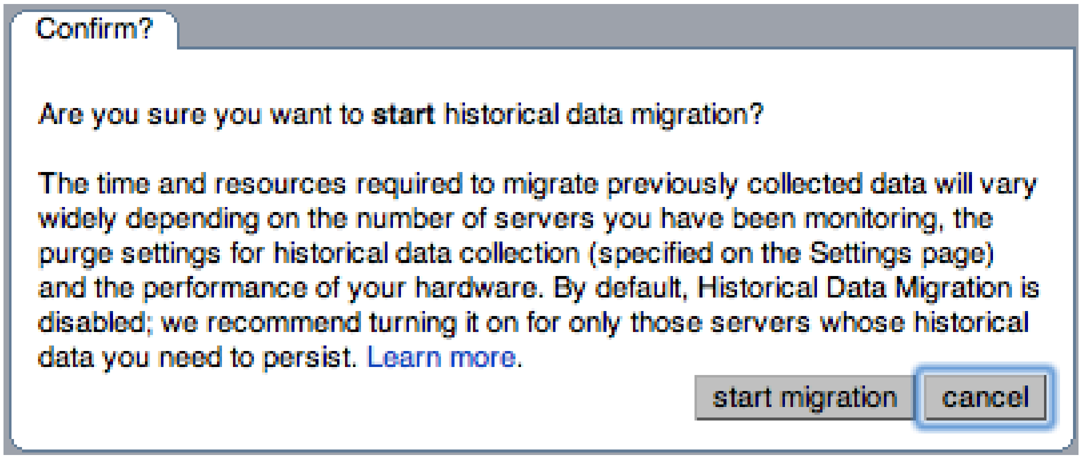
MySQL Enterprise © 2005-2008 MySQL AB, 2008 Sun Microsystems, Inc.. All rights reserved. Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "admin" (Nov 19, 2008 1:36:19 PM)
Monitoring 3 instances on 2 hosts (198 hosts remaining). Subscription is up-to-date. More info....

- Select the servers you want to migrate by using the checkbox next to each server name. You can select one or more servers to migrate. Servers that are suitable for migration will show their migration status within the **Migration Status** columnn. If the server is not able to be migrated, `N/A` will be shown.
- Click Start next to **Historical Data Migration**.
- You will be presented with a confirmation dialog box. To start the migration, click Start Migration. To cancel migration, click Cancel.

- The servers that have been selected for migration will show `Queued for Migration` in the **Migration Status** column.

Figure F.2. MySQL Enterprise Monitor: Confirming Historical Data Migration



Confirm?

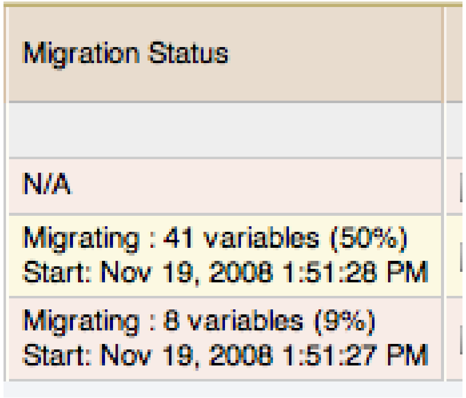
Are you sure you want to **start** historical data migration?

The time and resources required to migrate previously collected data will vary widely depending on the number of servers you have been monitoring, the purge settings for historical data collection (specified on the Settings page) and the performance of your hardware. By default, Historical Data Migration is disabled; we recommend turning it on for only those servers whose historical data you need to persist. Learn more.

Monitoring Historical Data Migration

You can check the migration status of any individual server by examining the **Migration Status** column for each server. You can see an example of the migration status below.

Figure F.3. MySQL Enterprise Monitor: Historical Data Migration Progress



Migration Status
N/A
Migrating : 41 variables (50%) Start: Nov 19, 2008 1:51:28 PM
Migrating : 8 variables (9%) Start: Nov 19, 2008 1:51:27 PM

Note that the migration status is shown according to the state of migration at the time the page was loaded. The actual migration continues in the background, and the current state may not match the state of the migration at the time it is viewed.

Servers showing `Done` in the **Migration Status** column have already completed their migration.

You can check the overall migration status by examining the **Upgrade Status** display.

Stopping Historical Data Migration

You can stop the migration process for any server that is still migrating data. The migration can be restarted at any time without causing any problems.

To stop the historical data migration:

1. Select the servers you want to stop migrating by using the checkbox next to each server name. You can select one or more servers

to stop migrating.

2. Click STOP next to **HISTORICAL DATA MIGRATION**.

Confirmation that the migration has been stopped will be provided. If migration has already completed, you will be notified.

Removing Old Data

Once data migration has been completed for all the servers you want to migrate, you may want to delete or remove access to the old data within your MySQL Enterprise Monitor repository. Data for MySQL Enterprise Monitor 1.3 was stored in a database called merlin within the MySQL repository. Data for MySQL Enterprise Monitor 2.0 is stored within a database called mem.

To create a backup of the old information, use mysqldump:

```
shell> mysqldump -uservice_manager -pPassword -P13306 -h127.0.0.1 merlin >data-1.3.sql
```

The above will create a file, data-1.3.sql containg all of the MySQL Enterprise Monitor 1.3 information.

If you remove access to the old data, then the data migration options for old servers will be removed from the **MANAGE SERVERS** panel within MySQL Enterprise Service Manager. To remove access, you need to REVOKE access to the merlin database:

```
mysql& REVOKE ALL on merlin.* FROM 'service_manager';
```

Note that revoking access to the old data will not reclaim any of the disk space used by the old data.

To delete the data from the database and free up the space being used by the historical information, DROP the merlin database:

```
mysql& DROP DATABASE merlin;
```

Once all the data has been migrated you can hide the migration user interface by clicking the HIDE MIGRATION INTERFACE button.

F.10. Regular MySQL Enterprise Monitor Maintenance

MySQL Enterprise Monitor is generally self managing and does not need excessive maintenance. You should, however, be aware of certain maintenance tasks that you can automate or will need to manually perform to keep your MySQL Enterprise Monitor running efficiently.

- Make sure you have set the purge interval for your data to an appropriate value according to duration and history of data that you want to keep. For more information, see Data Purge Behavior.
- Check, and delete, the contents of the temporary directory with your MySQL Enterprise Service Manager installation directory.

F.11. Advisor/Graph Reference

Basic

- **Advisors**
  - Connection Usage Excessive
  - CPU I/O Usage Excessive
  - CPU Usage Excessive
  - Lock Contention Excessive
  - MyISAM Key Cache Has Sub-Optimal Hit Rate
  - MySQL Agent Not Communicating With Database Server

- MySQL Agent Not Reachable
- MySQL Server Not Reachable
- Query Cache Has Sub-Optimal Hit Rate
- RAM Usage Excessive
- Table Scans Excessive
- Temporary Tables To Disk Ratio Excessive

- **Graphs**
  - Connections
  - CPU Utilization
  - Database Activity
  - Hit Ratios

Silver

- **Advisors**
  - 32-Bit Binary Running on 64-Bit AMD Or Intel System
  - Account Has An Overly Broad Host Specifier
  - Account Has Global Privileges
  - Account Has Old Insecure Password Hash
  - Account Has Strong MySQL Privileges
  - Attempted Connections To The Server Have Failed
  - Binary Logging Is Limited
  - Binary Logging Not Enabled
  - Binary Logging Not Synchronized To Disk At Each Write
  - CREATE TABLE LIKE Does Not Require Any Privileges On Source Table
  - Database May Not Be Portable Due To Identifier Case Sensitivity
  - Date-Handling Bugs Can Crash The Server
  - Default Value Being Used For max_prepared_stmt_count
  - Disabling Next-Key Locking In InnoDB Can Crash The Server
  - Excessive Percentage Of Attempted Connections To The Server Have Failed
  - General Query Log Enabled
  - Improper key_cache_block_size Setting Can Corrupt MyISAM Tables
  - In-Memory Temporary Table Size Limited By Maximum Heap Table Size
  - Incorrect InnoDB Flush Method On Windows

- InnoDB Redo Logs Not Sized Correctly
- InnoDB Tablespace Cannot Automatically Expand
- Insecure Password Authentication Option Is Enabled
- Insecure Password Generation Option Is Enabled
- Key Buffer Size Greater Than 4 GB
- LOCAL Option Of LOAD DATA Statement Is Enabled
- Malformed Password Packet In Connection Protocol Can Crash Server
- Maximum Connection Limit Nearing Or Reached
- Missing Security Improvements In GRANT Options
- Multi-Byte Encoding Processing Can Lead To SQL Injection
- Multiple Threads Used When Repairing MyISAM Tables
- MySQL Agent Memory Usage Excessive
- Next-Key Locking Disabled For InnoDB But Binary Logging Enabled
- No Limit On Total Number Of Prepared Statements
- No Value Set For myisam-recover
- Non-Authorized User Has DB, Table, Or Index Privileges On All Databases
- Non-Authorized User Has GRANT Privileges On All Databases
- Non-Authorized User Has Server Admin Privileges
- Query Cache Not Available
- Root Account Can Login Remotely
- Root Account Without Password
- Security Alterations Detected: User Privileges Granted
- Security Alterations Detected: User Privileges Revoked
- Security Alterations Have Been Detected
- Server Contains Default "test" Database
- Server Has Accounts Without A Password
- Server Has Anonymous Accounts
- Server Includes A Root User Account
- Stored Routine Runs In Definer''s Rather Than Caller''s Security Context
- Symlinks Are Enabled
- Table Cache Set Too Low For Startup
- Use Of View Overrides Column Update Privileges On Underlying Table
- User Can Gain Privileges By Running Stored Routine Declared Using SQL SECURITY INVOKER

- User Has Rights To Database That Does Not Exist
- User With Only ALTER Privilege On Partitioned Table Can Obtain SELECT Privilege Information
- Users Can View All Databases On MySQL Server
- XA Distributed Transaction Support Enabled For InnoDB

- **Graphs**
  - InnoDB Row Details
  - KBytes In/Out
  - Load Average
  - Memory Usage - Agent
  - Memory Usage - OS Resident
  - Row Accesses
  - Row Writes

Gold

- **Advisors**
  - Binary Log File Count Exceeds Specified Limit
  - Binary Log Space Exceeds Specified Limit
  - InnoDB Buffer Cache Has Sub-Optimal Hit Rate
  - INSERT ON DUPLICATE KEY UPDATE Bug May Break Replication
  - Key Buffer Size May Not Be Optimal For Key Cache
  - Key Buffer Size May Not Be Optimal For System RAM
  - Query Cache Potentially Undersized
  - Slave Detection Of Network Outages Too High
  - Slave Execution Position Too Far Behind Read Position
  - Slave Has Been Stopped
  - Slave Has Experienced A Replication Error
  - Slave Has Login Accounts With Inappropriate Privileges
  - Slave Has Problem Communicating With Master
  - Slave Has Stopped Replicating
  - Slave I/O Thread Not Running
  - Slave Not Configured As Read Only
  - Slave Relay Log Space Is Very Large
  - Slave Relay Logs Not Automatically Purged

- Slave SQL Thread Not Running
- Slave SQL Thread Reading From Older Relay Log Than I/O Thread
- Slave Too Far Behind Master
- Slave Waiting To Free Relay Log Space
- Slave Without REPLICATION SLAVE Accounts
- Table Cache Not Optimal
- Thread Cache Size May Not Be Optimal

- **Graphs**
  - InnoDB Buffer Pool
  - Memory Usage - OS Virtual
  - Opened Tables
  - Query Cache Blocks
  - Query Cache Memory
  - Replication Delay
  - Temporary Tables
  - Thread Cache

Platinum

- **Advisors**
  - Binary Log Usage Exceeding Disk Cache Memory Limits
  - Data Flushed To Disk After Each SQL Statement
  - Excessive Disk Temporary Table Usage Detected
  - Flush Time Set To Non-Zero Value
  - Indexes Not Being Used Efficiently
  - InnoDB Buffer Pool Writes May Be Performance Bottleneck
  - InnoDB Doublewrite Buffer Enabled
  - InnoDB Flush Method May Not Be Optimal
  - InnoDB Log Buffer Flushed To Disk After Each Transaction
  - InnoDB Log Waits May Be Performance Bottleneck
  - MyISAM Concurrent Insert Setting May Not Be Optimal
  - Object Changed: Database Has Been Altered
  - Object Changed: Database Has Been Created
  - Object Changed: Database Has Been Dropped

  - Object Changed: Function Has Been Created
  - Object Changed: Function Has Been Dropped
  - Object Changed: Index Has Been Created
  - Object Changed: Index Has Been Dropped
  - Object Changed: Table Has Been Altered
  - Object Changed: Table Has Been Created
  - Object Changed: Table Has Been Dropped
  - Object Changed: User Has Been Dropped
  - Object Changes Detected
  - Prepared Statements Not Being Closed
  - Prepared Statements Not Being Used Effectively
  - Query Cache Not Enabled
  - Server-Enforced Data Integrity Checking Disabled
  - Server-Enforced Data Integrity Checking Not Strict
  - Slow Query Log Not Enabled
  - Stored Procedures Found With SELECT * Syntax
  - Table Lock Contention Excessive
  - Thread Cache Not Enabled
  - Too Many Concurrent Queries Running
- **Graphs**
  - InnoDB OS File Access
  - InnoDB Semaphores
  - Query Cache Efficiency
  - Query Cache Lowmem Prunes
  - Query Cache Queries
  - Sort Activity
  - Table Locks

F.11.1. Advisors

Individual Advisor information is show below.

- **32-Bit Binary Running on 64-Bit AMD Or Intel System**

  The chip architecture and operating system installed on a machine both impact the performance of software running on the system. While it is possible to run 32-bit software on many 64-bit systems, in general, software built to run on a 64-bit system will run better on such a system than software built to run on a 32-bit system.

Default frequency 06:00:00

- **Account Has An Overly Broad Host Specifier**

  The MySQL server has user accounts with overly broad host specifiers. A MySQL account is identified by both a username and a hostname, which are found in the User and Host columns of the mysql.user table. The User value is the name that a client must supply when connecting to the server. The Host value indicates the host or hosts from which the user is allowed to connect. If this is a literal hostname, the account is limited to connections only from that host. If the hostname contains the '%' wildcard character, the user can connect from any host that matches the wildcard character and potentially from any host at all.

  From a security standpoint, literal host values are best and `%` is worst. Accounts that have Host values containing wildcards are more susceptible to attack than accounts with literal host values, because attackers can attempt to connect from a broader range of machines.

  For example, if an account has user and host values of `root` and `%` , it means that you can connect as the root user from `any` machine if you know the password. By contrast, if the host name is `localhost` or `127.0.0.1`, the attacker can only attempt to connect as the root user from the server host.

  Default frequency 00:05:00

- **Account Has Global Privileges**

  A MySQL server may have user accounts with privileges on all databases and tables (*.*). In most cases global privileges should be allowed only for the MySQL root user, and possibly for users that you trust or use for backup purposes. Global privileges such as `DROP, ALTER, DELETE, UPDATE, INSERT,` and `LOCK TABLES` may be dangerous as they may cause other users to be affected adversely.

  Default frequency 00:05:00

- **Account Has Old Insecure Password Hash**

  Prior to MySQL 4.1, password hashes computed by the PASSWORD() function were 16 bytes long. As of MySQL 4.1 (and later), PASSWORD() was modified to produce a longer 41-byte hash value to provide enhanced security.

  Default frequency 06:00:00

- **Account Has Strong MySQL Privileges**

  Certain account privileges can be dangerous and should only be granted to trusted users when necessary. For example, the FILE privilege allows a user to read and write files on the database server (which includes sensitive operating system files), the PROCESS privilege allows currently executing statements to be monitored, and the SHUTDOWN privilege allows a user to shut down the server. In addition, the GRANT privilege allows a user to grant privileges to others.

  Default frequency 00:05:00

- **Attempted Connections To The Server Have Failed**

  Aborted connection attempts to MySQL may indicate an issue with respect to the server or network, or could be indicative of DoS or password-cracking attempts against the MySQL Server. The aborted-connects count is incremented when:

  - A client does not have privileges to access a database
  - A client uses the wrong password
  - A malformed packet is received
  - The connect_timeout variable is exceeded

  Default frequency 00:05:00

- **Binary Log File Count Exceeds Specified Limit**

  The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables replication as well as point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database. However, binary logs consume disk space and file system resources, and can be removed from a production server after they are no longer needed by the slaves connecting to this master server, and after they have been backed up.

Default frequency 06:00:00

- **Binary Log Space Exceeds Specified Limit**

  The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables replication as well as point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database. However, binary logs consume disk space and can be removed from a production server after they are no longer needed by the slaves connecting to this master server, and after they have been backed up.

  Default frequency 06:00:00

- **Binary Log Usage Exceeding Disk Cache Memory Limits**

  When binary log usage exceeds the binary log cache memory limits, it is performing excessive disk operations. For optimal performance, transactions that move through the binary log should be contained within the binary log cache.

  Default frequency 00:05:00

- **Binary Logging Is Limited**

  The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

  Binary logging can be limited to specific databases with the `--binlog-do-db` and the `--binlog-ignore-db` options. However, if these options are used, your point-in-time recovery options are limited accordingly, along with your ability to review alterations made to your system.

  Default frequency 06:00:00

- **Binary Logging Not Enabled**

  The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

  Default frequency 06:00:00

- **Binary Logging Not Synchronized To Disk At Each Write**

  By default, the binary log is not synchronized to disk at each write. If the server host, operating system, or MySQL server crash, there is a chance that the latest statements in the binary log are not written to disk. To prevent this, you can cause the binary log to be synchronized to disk after every Nth binary log entry using the `sync_binlog` global variable. 1 is the safest value, but also the slowest.

  Default frequency 06:00:00

- **Connection Usage Excessive**

  Once the maximum connection limit for the MySQL server has been reached, no other user connections can be established and errors occur on the client side of the application.

  Default frequency 00:01:00

- **CPU I/O Usage Excessive**

  CPU I/O usage should be low on a properly configured and well-tuned system. Excessive CPU I/O usage is often indicative of poor disk or network performance.

  Default frequency 00:01:00

- **CPU Usage Excessive**

CPU usage should be low-to-moderate on a properly configured and well-tuned system. Excessive CPU usage can be indicative of many problems: insufficient RAM, fragmented disks, poorly-tuned queries, etc.

Default frequency 00:01:00

- **CREATE TABLE LIKE Does Not Require Any Privileges On Source Table**

  Due to Bug#25578, a user who does not have any access to a database can still clone the structure of tables in that database. Knowing the structure of tables in a database may give a determined hacker insight that allows him or her to proceed with other exploits.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **Data Flushed To Disk After Each SQL Statement**

  MySQL updates its data files on disk with the write() system call after every SQL statement and lets the operating system handle the synchronizing to disk. You can force MySQL to flush everything to disk after every SQL statement with the `--flush` option, however, this will have an adverse effect on performance.

  Default frequency 06:00:00

- **Database May Not Be Portable Due To Identifier Case Sensitivity**

  The case sensitivity of the underlying operating system determines the case sensitivity of database and table names. If you are using MySQL on only one platform, you don't normally have to worry about this. However, depending on how you have configured your server you may encounter difficulties if you want to transfer tables between platforms that differ in filesystem case sensitivity.

  Default frequency 06:00:00

- **Date-Handling Bugs Can Crash The Server**

  Two bugs related to date-handling operations can crash the server leading to potential Denial of Service (DoS) attacks:

  - STR_TO_DATE(1,NULL) caused a server crash (Bug#15828);
  - Invalid arguments to DATE_FORMAT() caused a server crash (Bug#20729).

    These bugs have been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **Default Value Being Used For max_prepared_stmt_count**

  Prepared statements may increase performance in applications that execute similar statements more than once, primarily because the query is parsed only once. Prepared statements can also reduce network traffic because it is only necessary to send the data for the parameters for each execution rather than the whole statement.

  However, prepared statements consume memory in the MySQL server until they are closed, so it is important to use them properly and to limit the number of statements that can be open at any one time. The default value for max_prepared_stmt_count may not be appropriate for your application and environment.

  Default frequency 06:00:00

- **Disabling Next-Key Locking In InnoDB Can Crash The Server**

  Due to several bugs, the server could crash if next-key locking in InnoDB was disabled.

  These bugs have been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **Excessive Disk Temporary Table Usage Detected**

  If the space required to build a temporary table exceeds either `tmp_table_size` or `max_heap_table_size`, MySQL creates a disk-based table in the server's tmpdir directory. Also, tables that have TEXT or BLOB columns are automatically placed on disk.

For performance reasons it is ideal to have most temporary tables created in memory, leaving exceedingly large temporary tables to be created on disk.

Default frequency 00:05:00

- **Excessive Percentage Of Attempted Connections To The Server Have Failed**

  Excess aborted connection attempts to MySQL may indicate an issue with respect to the server or network, or could be indicative of DoS or password-cracking attempts against the MySQL Server. The aborted-connects count is incremented when:

  - A client does not have privileges to access a database
  - A client uses the wrong password
  - A malformed packet is received
  - The connect_timeout variable is exceeded

  Default frequency 00:05:00

- **Flush Time Set To Non-Zero Value**

  If `flush_time` is set to a non-zero value, all tables are closed every flush_time seconds to free up resources and synchronize unflushed data to disk. If your system is unreliable and tends to lock up or restart often, forcing out table changes this way degrades performance but can reduce the chance of table corruption or data loss. We recommend that this option be used only on Windows, or on systems with minimal resources.

  Default frequency 06:00:00

- **General Query Log Enabled**

  The general query log is a general record of what mysqld is doing. The server writes information to this log when clients connect or disconnect, and it logs each SQL statement received from clients. The general query log can be very useful when you suspect an error in a client and want to know exactly what the client sent to mysqld.

  However, the general query log should not be enabled in production environments because:

  - It adds overhead to the server;
  - It logs statements in the order they were received, not the order they were executed, so it is not reliable for backup/recovery;
  - It grows fast and can use a lot of disk space;
  - You cannot stop logging to the general query log without stopping the server (for versions previous to 5.1).

    You should use the binary log instead.

  Default frequency 06:00:00

- **Improper key_cache_block_size Setting Can Corrupt MyISAM Tables**

  The server deducts some bytes from the key_cache_block_size option value and reduces it to the next lower 512 byte boundary. The resulting block size is not a power of two. Setting the key_cache_block_size system variable to a value that is not a power of two results in MyISAM table corruption.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **In-Memory Temporary Table Size Limited By Maximum Heap Table Size**

  If the space required to build a temporary table exceeds either `tmp_table_size` or `max_heap_table_size`, MySQL creates a disk-based table in the server's tmpdir directory. For performance reasons it is ideal to have most temporary tables created in memory, leaving exceedingly large temporary tables to be created on disk. Many DBAs configure `tmp_table_size` appropriately, but forget that `max_heap_table_size` also plays a role.

Default frequency 06:00:00

- **Incorrect InnoDB Flush Method On Windows**

  If `innodb_file_per_table` is enabled and `innodb_flush_method` is not set to `unbuffered` on Windows, MySQL may not start and you may see operating system error code 87.

  Default frequency 06:00:00

- **Indexes Not Being Used Efficiently**

  The target server does not appear to be using indexes efficiently. The values of Handler_read_rnd_next and Handler_read_rnd together - which reflect the number of rows read via full table scans - are high compared to the Handler variables which denote index accesses - such as Handler_read_key, Handler_read_next etc. You should examine your tables and queries for proper use of indexes.

  Default frequency 00:05:00

- **InnoDB Buffer Cache Has Sub-Optimal Hit Rate**

  Logical I/O is many times faster than physical I/O, and therefore a DBA should strive to keep physical I/O to a minimum. It is true that logical I/O is not free, and that the DBA should work to keep `all` I/O to a minimum, but it is best if most data access is performed in memory. When using InnoDB, most data access should occur in RAM, and therefore the InnoDB buffer cache hit rate should be high.

  Default frequency 00:05:00

- **InnoDB Buffer Pool Writes May Be Performance Bottleneck**

  For optimal performance, InnoDB should not have to wait before writing pages into the InnoDB buffer pool.

  Default frequency 00:05:00

- **InnoDB Doublewrite Buffer Enabled**

  InnoDB uses a novel file flush technique called *doublewrite*. It adds safety to recovery following an operating system crash or a power outage, and improves performance on most varieties of Unix by reducing the need for fsync() operations.

  Doublewrite means that before writing pages to a data file, InnoDB first writes them to a contiguous tablespace area called the doublewrite buffer. Only after the write and the flush to the doublewrite buffer has completed does InnoDB write the pages to their proper positions in the data file. If the operating system crashes in the middle of a page write, during recovery InnoDB can find a good copy of the page from the doublewrite buffer.

  Default frequency 06:00:00

- **InnoDB Flush Method May Not Be Optimal**

  Different values for `innodb_flush_method` can have a marked effect on InnoDB performance. In some versions of GNU/Linux and Unix, flushing files to disk by invoking fsync() (which InnoDB uses by default) or other similar methods, can be surprisingly slow. If you are dissatisfied with database write performance, you might try setting the innodb_flush_method parameter to O_DIRECT or O_DSYNC.

  Default frequency 06:00:00

- **InnoDB Log Buffer Flushed To Disk After Each Transaction**

  By default, InnoDB's log buffer is written out to the log file at each transaction commit and a flush-to-disk operation is performed on the log file, which enforces ACID compliance. In the event of a crash, if you can afford to lose a second's worth of transactions, you can achieve better performance by setting `innodb_flush_log_at_trx_commit` to either 0 or 2. If you set the value to 2, then only an operating system crash or a power outage can erase the last second of transactions. This can be very useful on slave servers, where the loss of a second's worth of data can be recovered from the master server if needed.

  Default frequency 06:00:00

- **InnoDB Log Waits May Be Performance Bottleneck**

For optimal performance, InnoDB should not have to wait before writing DML activity to the InnoDB log buffer.

Default frequency 00:05:00

- **InnoDB Redo Logs Not Sized Correctly**

  To avoid frequent checkpoint activity and reduce overall physical I/O, which can slow down write-heavy systems, the InnoDB redo logs should be approximately 50-100% of the size of the InnoDB buffer pool, depending on the size of the buffer pool.

  Default frequency 06:00:00

- **InnoDB Tablespace Cannot Automatically Expand**

  If the InnoDB tablespace is not allowed to automatically grow to meet incoming data demands and your application generates more data than there is room for, out-of-space errors will occur and your application may experience problems.

  Default frequency 06:00:00

- **Insecure Password Authentication Option Is Enabled**

  Prior to MySQL 4.1, password hashes computed by the PASSWORD() function were 16 bytes long. As of MySQL 4.1 (and later), PASSWORD() was modified to produce a longer 41-byte hash value to provide enhanced security. However, in order to allow backward-compatibility with user tables that have been migrated from pre-4.1 systems, you can configure MySQL to accept logins for accounts that have password hashes created using the old, less-secure PASSWORD() function, but this is not recommended.

  Default frequency 06:00:00

- **Insecure Password Generation Option Is Enabled**

  Prior to MySQL 4.1, password hashes computed by the PASSWORD() function were 16 bytes long. As of MySQL 4.1 (and later), PASSWORD() was modified to produce a longer 41-byte hash value to provide enhanced security. In order to allow backward-compatibility with older client programs, you can configure MySQL to generate short (pre-4.1) password hashes for new passwords, however, this is not recommended.

  Default frequency 06:00:00

- **INSERT ON DUPLICATE KEY UPDATE Bug May Break Replication**

  For INSERT ... ON DUPLICATE KEY UPDATE statements where some AUTO_INCREMENT values were generated automatically for inserts and some rows were updated, one auto-generated value was lost per updated row, leading to faster exhaustion of the range of the AUTO_INCREMENT column. Affected versions of MySQL include 5.0.24 to 5.0.34, and 5.1.12 to 5.1.17 (inclusive).

  Because the original problem can affect replication (different values on master and slave), it is recommended that the master and its slaves be upgraded to the current version.

  Default frequency 06:00:00

- **Key Buffer Size Greater Than 4 GB**

  To minimize disk I/O, the MyISAM storage engine employs a key cache (or key buffer) to keep the most frequently accessed index blocks in memory. However, prior to MySQL version 5.0.52 this key buffer is limited in size to 4 GB, `even on 64-bit operating systems`. If set to a larger value, mysqld may crash when it tries to increase the actual buffer beyond 4 GB. Note that key_buffer_size is limited to 4GB on both 32-bit and 64-bit Windows systems, even in MySQL version 5.0.52 and later.

  Default frequency 06:00:00

- **Key Buffer Size May Not Be Optimal For Key Cache**

  The key cache hit ratio represents the proportion of keys that are being read from the key cache in memory instead of from disk. This should normally be greater than 99% for optimum efficiency.

  Default frequency 00:05:00

- **Key Buffer Size May Not Be Optimal For System RAM**

The target server does not appear to have sufficient memory devoted to the key cache. On a dedicated server, this cache is commonly about 25%-50% of total RAM.

Default frequency 06:00:00

- **LOCAL Option Of LOAD DATA Statement Is Enabled**

  The LOAD DATA statement can load a file that is located on the server host, or it can load a file that is located on the client host when the LOCAL keyword is specified.

  There are two potential security issues with supporting the LOCAL version of LOAD DATA statements:

  - The transfer of the file from the client host to the server host is initiated by the MySQL server. In theory, a patched server could be built that would tell the client program to transfer a file of the server's choosing rather than the file named by the client in the LOAD DATA statement. Such a server could access any file on the client host to which the client user has read access.

  - In a Web environment where the clients are connecting from a separate web server, a user could use LOAD DATA LOCAL to read any files that the web server process has read access to (assuming that a user could run any statement against the SQL server). In this environment, the client with respect to the MySQL server actually is the web server, not the remote program being run by the user who connects to the web server.

  Default frequency 00:05:00

- **Lock Contention Excessive**

  Performance can be degraded if the percentage of table operations that have to wait for a lock is high compared to the overall number of locks. This can happen when using a table-level locking storage engine, such as MyISAM, instead of a row-level locking storage engine.

  Default frequency 00:05:00

- **Malformed Password Packet In Connection Protocol Can Crash Server**

  Due to Bug#28984, a malformed password packet in the connection protocol could cause the server to crash. This can lead to denial of service (DoS) attacks.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **Maximum Connection Limit Nearing Or Reached**

  Once the maximum connection limit for the MySQL server has been reached, no other user connections can be established and errors occur on the client side of the application.

  Default frequency 00:05:00

- **Missing Security Improvements In GRANT Options**

  The GRANT statement is used to create MySQL user accounts and to grant rights to accounts. Due to bugs 15756 and 14385, rights may be granted erroneously in certain circumstances:

  - In grant table comparisons, improper use of a latin1 collation caused some hostname matches to be true that should have been false (Bug#15756).

  - GRANTs to users with wildcards in their host information could be erroneously applied to similar users with the same username and similar wildcards. For example, a privilege granted to foo@% is also applied to user foo@192.% (Bug#14385).

    These bugs have been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **Multi-Byte Encoding Processing Can Lead To SQL Injection**

  Due to bug 8378, the server incorrectly parsed strings escaped with the mysql_real_escape_string() C API function. As a result, even when the character set-aware mysql_real_escape_string() function was used, SQL injection was possible.

This bug has been fixed in later versions of the MySQL server.

Default frequency 06:00:00

- **Multiple Threads Used When Repairing MyISAM Tables**

  Using multiple threads when repairing MyISAM tables can improve performance, but it can also lead to table and index corruption as reported by several bugs (#11527, #11684, #18874). Even though these bugs have been fixed, this feature is still considered beta-quality, as noted in the manual.

  Default frequency 06:00:00

- **MyISAM Concurrent Insert Setting May Not Be Optimal**

  MyISAM uses table-level locking, which can adversely affect performance when there are many concurrent INSERT and SELECT statements because INSERTs will block all SELECTs until the INSERT is completed. However, MyISAM can be configured to allow INSERT and SELECT statements to run concurrently in certain situations.

  - If `concurrent_insert is set to 1` (the default), MySQL allows INSERT and SELECT statements to run concurrently `for MyISAM tables that have no free blocks in the middle` of the data file.

  - If `concurrent_insert is set to 2` (available in MySQL 5.0.6 and later), MySQL allows concurrent inserts `for all MyISAM tables`, even those that have holes. For a table with a hole, new rows are inserted at the end of the table if it is in use by another thread. Otherwise, MySQL acquires a normal write lock and inserts the row into the hole.

    Note that setting concurrent_insert to 2 allows tables to grow even when there are holes in the middle. This can be bad for applications that delete large chunks of data but continue to issue many SELECTs, thus effectively preventing INSERTs from filling the holes.

  Default frequency 06:00:00

- **MyISAM Key Cache Has Sub-Optimal Hit Rate**

  The key cache hit ratio represents the proportion of index values that are being read from the key cache in memory instead of from disk. This should normally be greater than 99% for optimum efficiency.

  Default frequency 00:05:00

- **MySQL Agent Memory Usage Excessive**

  The memory needed by the MySQL Agent for basic monitoring is fairly small and consistent, and depends on the number of rules you have enabled. However, when the Query Analyzer is enabled, the Agent can use significantly more memory to monitor and analyze whatever queries you direct through it. In this case, the amount of memory used depends on the number of unique normalized queries, example queries and example explains being processed, plus the network bandwidth required to send query data to the Service Manager. In general, the amount of memory used for the Query Analyzer is small and well-bounded, but under some circumstances it can become excessive, especially on older versions of Linux.

  Default frequency 00:01:00

- **MySQL Agent Not Communicating With Database Server**

  The MySQL Enterprise Service Agent must be able to communicate with the local MySQL database server in order to monitor the server and provide advice on enforcement of best practices.

  Default frequency 00:01:00

- **MySQL Agent Not Reachable**

  In order to monitor a MySQL server, a Service Agent must be running and communicating with the Service Manager. If the Agent cannot communicate with the Service Manager, the Service Manager has no way of knowing if the MySQL database server being monitored is running, and it cannot collect current statistics to properly evaluate the rules scheduled against that server.

  Default frequency 00:00:01

- **MySQL Server Not Reachable**

  To perform useful work, it must be possible to connect to the local MySQL database server. If the MySQL Enterprise Service Agent cannot communicate with the server, it is likely the server is not running.

  Default frequency 00:01:00

- **Next-Key Locking Disabled For InnoDB But Binary Logging Enabled**

  Next-key locking in InnoDB can be disabled, which may improve performance in some situations. However, this may result in inconsistent data when recovering from the binary logs in replication or recovery situations. Starting from MySQL 5.0.2, this option is even more unsafe than it was in version 4.1.x.

  Default frequency 06:00:00

- **No Limit On Total Number Of Prepared Statements**

  Due to Bug#16365, there is no limit to the number of prepared statements that can be open per connection. This can lead to a Denial Of Service (DoS) attack, as the server will crash with out-of-memory (OOM) errors when the amount of statements becomes very large.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **No Value Set For myisam-recover**

  The `myisam-recover` option enables automatic MyISAM crash recovery should a MyISAM table become corrupt for some reason. If this option is not set, then a table will be "Marked as crashed" if it becomes corrupt, and no sessions will be able to SELECT from it, or perform any sort of DML against it.

  Default frequency 06:00:00

- **Non-Authorized User Has DB, Table, Or Index Privileges On All Databases**

  Privileges such as SELECT, INSERT, ALTER, and so forth allow a user to view and change data, as well as impact system performance. Such operations should be limited to only those databases to which a user truly needs such access so the user cannot inadvertently affect other people's applications and data stores.

  Default frequency 01:00:00

- **Non-Authorized User Has GRANT Privileges On All Databases**

  The `GRANT` privilege, when given on all databases as opposed to being limited to a few specific databases, enables a user to give to other users those privileges that the grantor possesses on all databases. It can be used for databases, tables, and stored routines. Such a privilege should be limited to as few users as possible. Users who do indeed need the GRANT privilege should have that privilege limited to only those databases they are responsible for, and not for all databases.

  Default frequency 01:00:00

- **Non-Authorized User Has Server Admin Privileges**

  Certain privileges, such as SHUTDOWN and SUPER, are primarily used for server administration. Some of these privileges can have a dramatic effect on a system because they allow someone to shutdown the server or kill running processes. Such operations should be limited to a small set of users.

  Default frequency 01:00:00

- **Object Changed: Database Has Been Altered**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Database Has Been Created**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Database Has Been Dropped**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Function Has Been Created**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Function Has Been Dropped**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures or functions and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Index Has Been Created**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Index Has Been Dropped**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Table Has Been Altered**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Table Has Been Created**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changed: Table Has Been Dropped**

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

Default frequency 00:10:00

- **Object Changed: User Has Been Dropped**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Object Changes Detected**

  For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

  Default frequency 00:10:00

- **Prepared Statements Not Being Closed**

  Prepared statements may increase performance in applications that execute similar statements more than once, primarily because the query is parsed only once. Prepared statements can also reduce network traffic because it is only necessary to send the data for the parameters for each execution rather than the whole statement.

  However, prepared statements take time to prepare and consume memory in the MySQL server until they are closed, so it is important to use them properly. If you are not closing prepared statements when you are done with them, you are needlessly tying up memory that could be put to use in other ways.

  Default frequency 00:05:00

- **Prepared Statements Not Being Used Effectively**

  Prepared statements may increase performance in applications that execute similar statements more than once, primarily because the query is parsed only once. Prepared statements can also reduce network traffic because it is only necessary to send the data for the parameters for each execution rather than the whole statement.

  However, prepared statements take time to prepare and consume memory in the MySQL server until they are closed, so it is important to use them properly. If you are only executing a statement a few times, the overhead of creating a prepared statement may not be worthwhile.

  Default frequency 00:05:00

- **Query Cache Has Sub-Optimal Hit Rate**

  When enabled, the query cache should experience a high degree of "hits", meaning that queries in the cache are being reused by other user connections. A low hit rate may mean that not enough memory is allocated to the cache, identical queries are not being issued repeatedly to the server, or that the statements in the query cache are invalidated too frequently by INSERT, UPDATE or DELETE statements.

  Default frequency 00:05:00

- **Query Cache Not Available**

  MySQL can cache the results of SELECT statements in memory so that they do not have to constantly be parsed and executed. If your application often runs the same queries over and over, caching the results can increase performance significantly. It's important to use a version or binary of MySQL that supports the query cache.

  Default frequency 06:00:00

- **Query Cache Not Enabled**

Enabling the query cache can increase performance by 200% for queries that are executed often and have large result sets.

Default frequency 00:05:00

- **Query Cache Potentially Undersized**

  When the Query Cache is full, and needs to add more queries to the cache, it will make more room in the cache by freeing the least recently used queries from the cache, and then inserting the new queries. If this is happening often then you should increase the size of the cache to avoid this constant "swapping".

  Default frequency 00:05:00

- **RAM Usage Excessive**

  A reasonable amount of free memory is required for a system to perform well. Without free memory, new processes and threads cannot start, and the operating system may do excessive paging (swapping blocks of memory to and from disk).

  Default frequency 00:01:00

- **Root Account Can Login Remotely**

  By default, MySQL includes a root account with unlimited privileges that is typically used to administer the MySQL server. If possible, accounts with this much power should not allow remote logins in order to limit access to only those users able to login to the machine on which MySQL is running. This helps prevent unauthorized users from accessing and changing the system.

  Default frequency 00:05:00

- **Root Account Without Password**

  The root user account has unlimited privileges and is intended for administrative tasks. Privileged accounts should have strong passwords to prevent unauthorized users from accessing and changing the system.

  Default frequency 00:05:00

- **Security Alterations Detected: User Privileges Granted**

  For development environments, changes to database security privileges may be a normal occurrence, but for production environments it is wise to know when any security changes occur with respect to database privileges, and to ensure that those changes are authorized and required.

  Default frequency 00:05:00

- **Security Alterations Detected: User Privileges Revoked**

  For development environments, changes to database security privileges may be a normal occurrence, but for production environments it is wise to know when any security changes occur with respect to database privileges, and to ensure that those changes are authorized and required.

  Default frequency 00:05:00

- **Security Alterations Have Been Detected**

  For development environments, changes to database security privileges may be a normal occurrence, but for production environments it is wise to know when any security changes occur with respect to database privileges, and to ensure that those changes are authorized and required.

  Default frequency 00:05:00

- **Server Contains Default "test" Database**

  By default, MySQL comes with a database named `test` that anyone can access. This database is intended only for testing and should be removed before moving into a production environment. Because the default `test` database can be accessed by any user and has permissive privileges, it should be dropped immediately as part of the installation process.

  Default frequency 00:05:00

- **Server Has Accounts Without A Password**

  Accounts without passwords are particularly dangerous because an attacker needs to guess only a username. Assigning passwords to all accounts helps prevent unauthorized users from accessing the system.

  Default frequency 00:05:00

- **Server Has Anonymous Accounts**

  Anonymous MySQL accounts allow clients to connect to the server without specifying a username. Since anonymous accounts are well known in MySQL, removing them helps prevent unauthorized users from accessing the system.

  Default frequency 00:05:00

- **Server Includes A Root User Account**

  By default, MySQL includes a root account with unlimited privileges that is typically used to administer the MySQL server. There is no reason this account must be named root. Accounts with this much power should not be easily discovered. Since the root account is well known in MySQL, changing its name helps prevent unauthorized users from accessing and changing the system.

  Default frequency 00:05:00

- **Server-Enforced Data Integrity Checking Disabled**

  SQL Modes define what SQL syntax MySQL should support and what kind of data validation checks it should perform. If no SQL modes are enabled this means there is no form of server-enforced data integrity, which means incoming data that is invalid will not be rejected by the server, but instead will be changed to conform to the target column's default datatype. Note, however, that beginning with MySQL 4.1, any client can change its own session SQL mode value at any time.

  Default frequency 06:00:00

- **Server-Enforced Data Integrity Checking Not Strict**

  SQL Modes define what SQL syntax MySQL should support and what kind of data validation checks it should perform. There are many possible options that can be used in conjunction with each other to specify varying degrees of syntax and data validation checks the MySQL server will perform. However, to ensure the highest level of confidence for data integrity, at least one of the following should be included in the list: `TRADITIONAL, STRICT_TRANS_TABLES,` or `STRICT_ALL_TABLES`.

  Note, however, that beginning with MySQL 4.1, any client can change its own session SQL mode value at any time.

  Default frequency 06:00:00

- **Slave Detection Of Network Outages Too High**

  Slaves must deal with network connectivity outages that affect the ability of the slave to get the latest data from the master, and hence cause replication to fall behind. However, the slave notices the network outage only after receiving no data from the master for `slave_net_timeout seconds`. You may want to decrease `slave_net_timeout` so the outages -- and associated connection retries -- are detected and resolved faster. The default for this parameter is 3600 seconds (1 hour), which is too high for many environments.

  Default frequency 06:00:00

- **Slave Execution Position Too Far Behind Read Position**

  When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. If the position from which the SQL thread is reading is way behind the position to which the I/O thread is currently writing, it is a sign that replication is getting behind and results of queries directed to the slave may not reflect the latest changes made on the master.

  Default frequency 00:05:00

- **Slave Has Been Stopped**

  If replication on a slave has been stopped, it means the slave is not retrieving the latest statements from the master and it is not executing those statements on the slave.

Default frequency 00:01:00

- **Slave Has Experienced A Replication Error**

  When a slave receives updates from its master it must apply those updates locally so the data on the slave matches that on the server. If an error occurs while applying an update on a slave, the data on the slave may not match that on the master and it is an indication that replication may be broken.

  Default frequency 00:05:00

- **Slave Has Login Accounts With Inappropriate Privileges**

  Altering and dropping tables on a slave can break replication. Unless the slave also hosts non-replicated tables, there is no need for accounts with these privileges.

  Default frequency 06:00:00

- **Slave Has Problem Communicating With Master**

  Slaves must connect to a master to get the latest data from the master. If they cannot connect, or periodically have trouble connecting, replication may fall behind (i.e. the slave may not have the lastest data that was written to the master).

  Default frequency 00:05:00

- **Slave Has Stopped Replicating**

  If neither the slave I/O thread nor the slave SQL threads are running, it means the slave is not getting the latest statements from the master and it is not executing those statements on the slave, and thus replication has stopped entirely.

  Default frequency 00:01:00

- **Slave I/O Thread Not Running**

  The slave I/O thread is the thread that retrieves statements from the master's binary log and records them into the slave's relay log. If this thread isn't running, it means the slave is not able to retrieve the latest data from the master.

  Default frequency 00:01:00

- **Slave Not Configured As Read Only**

  Arbitrary or unintended updates to a slave may break replication or cause a slave to be inconsistent with respect to its master. Making a slave `read_only` can be useful to ensure that a slave accepts updates only from its master server and not from clients; it minimizes the possibility of unintended updates.

  Default frequency 06:00:00

- **Slave Relay Log Space Is Very Large**

  When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. After the SQL thread has executed all the updates in a relay log, the file is no longer needed and can be deleted to conserve disk space.

  Default frequency 06:00:00

- **Slave Relay Logs Not Automatically Purged**

  When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. After the SQL thread has executed all the updates in a relay log, the file is no longer needed and can be deleted to conserve disk space.

  Default frequency 06:00:00

- **Slave SQL Thread Not Running**

  The slave SQL thread is the thread that reads statements from the slave's relay log and executes them to bring the slave in sync with the master. If this thread isn't running, it means the slave is not able to apply the latest changes it has read from the master, and results of queries directed to the slave may not reflect the latest changes made on the master.

Default frequency 00:01:00

- **Slave SQL Thread Reading From Older Relay Log Than I/O Thread**

  When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. If the SQL thread is reading from an older relay log than the one to which the I/O thread is currently writing, it is a sign that replication is getting behind and results of queries directed to the slave may not reflect the latest changes made on the master.

  Default frequency 00:05:00

- **Slave Too Far Behind Master**

  If a slave is too far behind the master, results of queries directed to the slave may not reflect the latest changes made on the master.

  Default frequency 00:01:00

- **Slave Waiting To Free Relay Log Space**

  For slaves with limited disk space you can place a limit on how large the replication relay log can grow. When the limit is reached, the I/O thread stops reading binary log events from the master server until the SQL thread has caught up and deleted some unprocessed relay logs. While this protects MySQL from filling up the disk, it means replication is delayed and the slave will fall behind the master.

  Default frequency 00:05:00

- **Slave Without REPLICATION SLAVE Accounts**

  If the master ever fails, you may want to use one of the slaves as the new master. An account with the REPLICATION SLAVE privilege must exist for a server to act as a replication master (so a slave can connect to it), so it's a good idea to create this account on your slaves to prepare it to take over for a master if needed.

  Default frequency 06:00:00

- **Slow Query Log Not Enabled**

  The slow query log can be used to identify queries that take a long time to complete.

  Default frequency 00:05:00

- **Stored Procedures Found With SELECT * Syntax**

  Best practices for SQL coding state that no query should be issued with SELECT *. Reasons include:

  - To ensure that only the necessary columns are returned from a SQL statement, the actual column names should be specifically entered. This cuts down on unwanted network traffic as only columns necessary for query satisfaction are present.

  - If the underlying table has columns added or removed, the query itself may malfunction if cursors or other such application objects are used.

  Default frequency 06:00:00

- **Stored Routine Runs In Definer''s Rather Than Caller''s Security Context**

  Due to bug 18630, a stored routine created by one user and then made accessible to a different user using GRANT EXECUTE could be executed by that user with the privileges of the routine's definer.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **Symlinks Are Enabled**

You can move tables and databases from the database directory to other locations and replace them with symbolic links to the new locations. You might want to do this, for example, to move a database to a file system with more free space or to increase the speed of your system by spreading your tables to different disks.

However, symlinks can compromise security. This is especially important if you run mysqld as root, because anyone who has write access to the server's data directory could then delete any file in the system!

Default frequency 06:00:00

- **Table Cache Not Optimal**

  MySQL is multi-threaded, so there may be many clients issuing queries for a given table simultaneously. To minimize the problem with multiple client threads having different states on the same table, the table is opened independently by each concurrent thread.

  The table cache is used to cache file descriptors for open tables and there is a single cache shared by all clients. Increasing the size of the table cache allows mysqld to keep more tables open simultanously by reducing the number of file open and close operations that must be done. If the value of `Open_tables` is approaching the value of `table_cache`, this may indicate performance problems.

  Default frequency 00:05:00

- **Table Cache Set Too Low For Startup**

  The table cache size controls the number of open tables that can occur at any one time on the server. MySQL will work to open and close tables as needed, however you should avoid having the table cache set too low, causing MySQL to constantly open and close tables to satisfy object access.

  If the table cache limit has been exceeded by the number of tables opened in the first three hours of service, then the table cache size is likely set too low.

  Default frequency 00:30:00

- **Table Lock Contention Excessive**

  Performance can be degraded if the percentage of table operations that have to wait for a lock is high compared to the overall number of locks. This can happen when using a table-level locking storage engine, such as MyISAM, instead of a row-level locking storage engine.

  Default frequency 00:05:00

- **Table Scans Excessive**

  The target server does not appear to be using indexes efficiently. The values of Handler_read_rnd_next and Handler_read_rnd together - which reflect the number of rows read via full table scans - are high compared to the sum of Handler variables which denote all row accesses - such as Handler_read_key, Handler_read_next etc. You should examine your tables and queries for proper use of indexes.

  Default frequency 00:05:00

- **Temporary Tables To Disk Ratio Excessive**

  If the space required to build a temporary table exceeds either `tmp_table_size` or `max_heap_table_size`, MySQL creates a disk-based table in the server's tmpdir directory. Also, tables that have TEXT or BLOB columns are automatically placed on disk.

  For performance reasons it is ideal to have most temporary tables created in memory, leaving exceedingly large temporary tables to be created on disk.

  Default frequency 00:05:00

- **Thread Cache Not Enabled**

  Each connection to the MySQL database server runs in its own thread. Thread creation takes time, so rather than killing the thread when a connection is closed, the server can keep the thread in its thread cache and use it for a new connection later.

  Default frequency 00:05:00

- **Thread Cache Size May Not Be Optimal**

  Each connection to the MySQL database server runs in its own thread. Thread creation takes time, so rather than killing the thread when a connection is closed, the server can keep the thread in its thread cache and use it for a new connection later.

  Default frequency 00:05:00

- **Too Many Concurrent Queries Running**

  Too many active queries indicates there is a severe load on the server, and may be a sign of lock contention or unoptimized SQL queries.

  Default frequency 00:05:00

- **Use Of View Overrides Column Update Privileges On Underlying Table**

  Due to Bug#27878, by using a view, a user who only has privileges to update a given column of a table is able to update any column of that table, even though the view is defined with SQL SECURITY INVOKER. Also, use of a view could allow a user to gain update privileges for tables in other databases.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **User Can Gain Privileges By Running Stored Routine Declared Using SQL SECURITY INVOKER**

  Due to Bug#27337, if a stored routine was declared using SQL SECURITY INVOKER, a user who invoked the routine could gain privileges. For example, a user without the CREATE privilege on a certain database could gain that privilege after invoking a stored routine.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **User Has Rights To Database That Does Not Exist**

  When a database is dropped, user privileges on the database are not automatically dropped. This has security implications as that user will regain privileges if a database with the same name is created in the future, which may not be the intended result.

  Default frequency 00:05:00

- **User With Only ALTER Privilege On Partitioned Table Can Obtain SELECT Privilege Information**

  Due to Bug#23675, a user with only the ALTER privilege on a partitioned table could obtain information about the table that should require the SELECT privilege.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

- **Users Can View All Databases On MySQL Server**

  The SHOW DATABASES privilege should be granted only to users who need to see all the databases on a MySQL Server. It is recommended that the MySQL Server be started with the `--skip-show-database` option enabled to prevent anyone from using the SHOW DATABASES statement unless they have been specifically granted the SHOW DATABASES privilege.

  Note: If a user is granted any global privilege, such as CREATE TEMPORARY TABLES or LOCK TABLES, they are automatically given the ability to show databases unless the server is started with the --skip-show-database option enabled. DBAs should be aware of this fact, in the event that any applications make use of temporary tables.

  Default frequency 00:05:00

- **XA Distributed Transaction Support Enabled For InnoDB**

  XA Distributed Transaction support is turned on by default. If you are not using this feature, note that it adds an extra fsync for each transaction and may adversely affect performance.

Default frequency 06:00:00

F.11.2. Graph Definition Reference

Individual graphs are defined below.

Connections

Name	Connections
Range Label	total/min
Series	**Expression**
Aborted	`aborted`
Active	`running`
Current	`connected`
Variables	**Data Item**
connected	`mysql:status:Threads_connected`
running	`mysql:status:Threads_running`
aborted	`mysql:status:Aborted_connects`

CPU Utilization

Name	CPU Utilization
Range Label	%
Series	**Expression**
Kernel	`sys/(sys+user+wait+idle)*100`
Total	`(sys+user+wait)/(sys+user+wait+idle)*100`
User	`user/(sys+user+wait+idle)*100`
Wait I/O	`wait/(sys+user+wait+idle)*100`
Variables	**Data Item**
sys	`os:cpu:cpu_sys`
user	`os:cpu:cpu_user`
wait	`os:cpu:cpu_wait`
idle	`os:cpu:cpu_idle`

Database Activity

Name	Database Activity
Range Label	avg statements/sec
Series	**Expression**
Call	`((call_proc)/60)`
Delete	`((deletes+delete_multi)/60)`
Insert	`((inserts+insert_selects)/60)`
Replace	`((replaces+replace_selects)/60)`
Select	`(selects/60)`
Update	`((updates+update_multi)/60)`
Variables	**Data Item**
selects	`mysql:status:Com_select`

inserts	mysql:status:Com_insert
insert_selects	mysql:status:Com_insert_select
updates	mysql:status:Com_update
update_multi	mysql:status:Com_update_multi
replaces	mysql:status:Com_replace
replace_selects	mysql:status:Com_replace_select
deletes	mysql:status:Com_delete
delete_multi	mysql:status:Com_delete_multi
call_proc	mysql:status:Com_call_procedure

Hit Ratios

Name	Hit Ratios
Range Label	%
Series	**Expression**
InnoDB Buffer	100-((iReads / iReadRequests)*100)
Key Cache	(100-((keyReads / keyReadRequests)*100)) * (keyReadRequests >= keyReads)
Query Cache	(qHits/(qHits+qInserts))*100
Thread Cache	100-((tcreated / connections)*100)
Variables	**Data Item**
qHits	mysql:status:Qcache_hits
qInserts	mysql:status:Qcache_inserts
keyReads	mysql:status:Key_reads
keyReadRequests	mysql:status:Key_read_requests
iReads	mysql:status:Innodb_buffer_pool_reads
iReadRequests	mysql:status:Innodb_buffer_pool_read_requests
tcreated	mysql:status:Threads_created
connections	mysql:status:Connections

InnoDB Buffer Pool

Name	InnoDB Buffer Pool
Range Label	MB
Series	**Expression**
Modified	(modified*16384)/(1024*1024)
Total Size	(size*16384)/(1024*1024)
Used	(used*16384)/(1024*1024)
Variables	**Data Item**
modified	mysql:innodbstatus:innodb_bp_modified_pages
size	mysql:innodbstatus:innodb_bp_size
used	mysql:innodbstatus:innodb_bp_db_pages

InnoDB OS File Access

Name	InnoDB OS File Access

Range Label	avg operations/sec
Series	**Expression**
File fsync()	`(fsync/60)`
File Reads	`(read/60)`
File Writes	`(write/60)`
Variables	**Data Item**
read	`mysql:innodbstatus:innodb_io_os_file_reads`
write	`mysql:innodbstatus:innodb_io_os_file_writes`
fsync	`mysql:innodbstatus:innodb_io_os_file_fsyncs`

InnoDB Row Details

Name	InnoDB Row Details
Range Label	avg rows/sec
Series	**Expression**
Rows Deleted	`(deleted/60)`
Rows Inserted	`(inserted/60)`
Rows Read	`(read/60)`
Rows Updated	`(udpated/60)`
Variables	**Data Item**
read	`mysql:innodbstatus:innodb_rows_read`
inserted	`mysql:innodbstatus:innodb_rows_inserted`
updated	`mysql:innodbstatus:innodb_rows_updated`
deleted	`mysql:innodbstatus:innodb_rows_deleted`

InnoDB Semaphores

Name	InnoDB Semaphores
Range Label	avg waits/sec
Series	**Expression**
OS Waits	`(oswaits/60)`
Spin Rounds	`(srounds/60)`
Spin Waits	`(swaits/60)`
Variables	**Data Item**
swaits	`mysql:innodbstatus:innodb_sem_mutex_spin_waits`
srounds	`mysql:innodbstatus:innodb_sem_mutex_rounds`
oswaits	`mysql:innodbstatus:innodb_sem_mutex_os_waits`

KBytes In/Out

Name	KBytes In/Out
Range Label	avg kbytes/sec
Series	**Expression**
Received	`((bytesIn/1024)/60)`
Sent	`((bytesOut/1024)/60)`
Total	`(((bytesIn+bytesOut)/1024)/60)`

Variables	Data Item
bytesIn	mysql:status:Bytes_received
bytesOut	mysql:status:Bytes_sent

Load Average

Name	Load Average
Range Label	Load Average
Series	Expression
1	zero
15	two
5	one
Variables	Data Item
zero	os:loadavg:0
one	os:loadavg:1
two	os:loadavg:2

Memory Usage - Agent

Name	Memory Usage - Agent
Range Label	MB
Series	Expression
Agent	agent_mem_size / 1024 / 1024
Lua	lua_mem_size / 1024 / 1024
Variables	Data Item
lua_mem_size	agent:lua:mem_size
agent_mem_size	agent:proc:mem_resident

Memory Usage - OS Resident

Name	Memory Usage - OS Resident
Range Label	MB
Series	Expression
Total	ram_total/(1024*1024)
Used	(ram_total-ram_unused)/(1024*1024)
Variables	Data Item
ram_total	os:mem:ram_total
ram_unused	os:mem:ram_unused

Memory Usage - OS Virtual

Name	Memory Usage - OS Virtual
Range Label	MB
Series	Expression
Total	swap_total/(1024*1024)
Used	(swap_total-swap_unused)/(1024*1024)

Variables	Data Item
swap_total	os:mem:swap_total
swap_unused	os:mem:swap_unused

Opened Tables

Name	Opened Tables
Range Label	total/min
Series	Expression
Opened Tables	openedTables
Variables	Data Item
openedTables	mysql:status:Opened_tables

Query Cache Blocks

Name	Query Cache Blocks
Range Label	num blocks
Series	Expression
Free	free_blocks
Size	size_blocks
Variables	Data Item
size_blocks	mysql:status:Qcache_total_blocks
free_blocks	mysql:status:Qcache_free_blocks

Query Cache Efficiency

Name	Query Cache Efficiency
Range Label	avg cache ops/sec
Series	Expression
Hits	(hits/60)
Inserts	(inserts/60)
Not Cached	(not_cached/60)
Variables	Data Item
hits	mysql:status:Qcache_hits
inserts	mysql:status:Qcache_inserts
not_cached	mysql:status:Qcache_not_cached

Query Cache Lowmem Prunes

Name	Query Cache Lowmem Prunes
Range Label	avg cache ops/sec
Series	Expression
Lowmem Prunes	(deletes/60)
Variables	Data Item
deletes	mysql:status:Qcache_lowmem_prunes

Query Cache Memory

Name	Query Cache Memory
Range Label	MB
Series	**Expression**
Free MB	`free/(1024*1024)`
Size MB	`size/(1024*1024)`
Variables	**Data Item**
size	`mysql`:`variables`:`query_cache_size`
free	`mysql`:`status`:`Qcache_free_memory`

Query Cache Queries

Name	Query Cache Queries
Range Label	num queries
Series	**Expression**
Queries in Cache	`queries`
Variables	**Data Item**
queries	`mysql`:`status`:`Qcache_queries_in_cache`

Replication Delay

Name	Replication Delay
Range Label	total seconds
Series	**Expression**
Seconds Behind Master	`sbehind`
Variables	**Data Item**
sbehind	`mysql`:`slavestatus`:`Seconds_Behind_Master`

Row Accesses

Name	Row Accesses
Range Label	avg rows/sec
Series	**Expression**
Average Rows Per Query	`((first+key+next+prev+hread_rnd+hread_rnd_next+sort_rows) / questions)`
Rows Read via Full Scan	`((hread_rnd+hread_rnd_next)/60)`
Rows Read via Indexes	`((first+key+next+prev)/60)`
Variables	**Data Item**
first	`mysql`:`status`:`Handler_read_first`
key	`mysql`:`status`:`Handler_read_key`
next	`mysql`:`status`:`Handler_read_next`
prev	`mysql`:`status`:`Handler_read_prev`
hread_rnd	`mysql`:`status`:`Handler_read_rnd`
hread_rnd_next	`mysql`:`status`:`Handler_read_rnd_next`
sort_rows	`mysql`:`status`:`Sort_rows`
questions	`mysql`:`status`:`Questions`

Row Writes

Name	Row Writes
Range Label	avg rows/sec
Series	**Expression**
Rows Deleted	`(delete/60)`
Rows Inserted	`(write/60)`
Rows Updated	`(update/60)`
Variables	**Data Item**
delete	`mysql`:`status`:`Handler_delete`
update	`mysql`:`status`:`Handler_update`
write	`mysql`:`status`:`Handler_write`

Sort Activity

Name	Sort Activity
Range Label	total/min
Series	**Expression**
Merge Passes	`Sort_merge_passes`
Range	`Sort_range`
Scan	`Sort_scan`
Variables	**Data Item**
Sort_merge_passes	`mysql`:`status`:`Sort_merge_passes`
Sort_range	`mysql`:`status`:`Sort_range`
Sort_scan	`mysql`:`status`:`Sort_scan`

Table Locks

Name	Table Locks
Range Label	avg locks/sec
Series	**Expression**
Immediate	`(locks_immediate/60)`
Waited	`(locks_waited/60)`
Variables	**Data Item**
locks_waited	`mysql`:`status`:`Table_locks_waited`
locks_immediate	`mysql`:`status`:`Table_locks_immediate`

Temporary Tables

Name	Temporary Tables
Range Label	total/min
Series	**Expression**
Disk Temp Tables	`diskTempTables`
Memory Temp Tables	`memoryTempTables`
Variables	**Data Item**
memoryTempTables	`mysql`:`status`:`Created_tmp_tables`
diskTempTables	`mysql`:`status`:`Created_tmp_disk_tables`

Thread Cache

Name	Thread Cache
Range Label	total/min
Series	**Expression**
New Connections	`connections`
Threads Created	`tcreated`
Variables	**Data Item**
tcreated	`mysql`:`status`:`Threads_created`
connections	`mysql`:`status`:`Connections`

Appendix G. Data Collection Items

Note

MySQL Enterprise subscription, MySQL Enterprise Monitor, MySQL Replication Monitor, and MySQL Query Analyzer are only available to commercial customers. To learn more, see: http://www.mysql.com/products/enterprise/features.html.

This appendix documents the data collection items used to create rules. These items are listed in the order that they appear in the **DATA ITEM** drop-down list box when creating a rule definition. For more information about creating and editing rules see Section 5.3, “Editing Built-in Rules” and Section 5.4.2, “Overview of Rule Creation”.

Table G.1. MySQL Monitor Data Items

Namespace	Namespace Type	Attribute	Type	Description
monitor	resourceBundle	`subscription`	`string`	
monitor	resourceBundle	`version`	`string`	
mysql	account_old_password	`user`	`string`	
mysql	active_count	`active_count`	`long`	
mysql	Agent	`agent.reachable`	`string`	
mysql	Agent	`host_id`	`string`	
mysql	Agent	`name`	`string`	
mysql	Agent	`version`	`string`	
mysql	anonymous_user	`user_count`	`long`	
mysql	avail_count	`avail_count`	`long`	
mysql	broad_host_specifier	`user`	`string`	
mysql	column	`Default`	`string`	The default value of the column
mysql	column	`Extra`	`string`	Any additional information about the column
mysql	column	`Field`	`string`	The name of the column
mysql	column	`Key`	`string`	Whether the column is indexed
mysql	column	`Null`	`string`	Whether NULL is allowed
mysql	column	`Type`	`string`	The data type of the column
mysql	committed_count	`committed_count`	`long`	
mysql	Database	`Database`	`string`	
mysql	Database	`name`	`string`	
mysql	Explain	`extra`	`string`	
mysql	Explain	`id`	`long`	
mysql	Explain	`key`	`string`	
mysql	Explain	`key_len`	`string`	
mysql	Explain	`possible_keys`	`string`	
mysql	Explain	`ref`	`string`	
mysql	Explain	`rows`	`long`	
mysql	Explain	`select_type`	`string`	
mysql	Explain	`table`	`string`	
mysql	Explain	`type`	`string`	
mysql	falcon_database_io	`logical_reads`	`long`	
mysql	falcon_database_io	`physical_reads`	`long`	
mysql	falcon_record_cache_summ	`total_recordcache_mb`	`double`	

Namespace	Namespace Type	Attribute	Type	Description
	ary			
mysql	falcon_record_cache_summary	total_space_mb	double	
mysql	falcon_system_memory_summary	free_system_mb	double	
mysql	falcon_system_memory_summary	total_system_mb	double	
mysql	falcon_transaction_summary	committed_txns	long	
mysql	falcon_transaction_summary	rolled_back_txns	long	
mysql	global_privileges	user	string	
mysql	grant_privileges	user_spec	string	
mysql	inappropriate_slave_privileges	user	string	
mysql	index	Cardinality	long	An estimate of the number of unique values in the index
mysql	index	Collation	string	How the column is sorted in the index
mysql	index	Column_name	string	The column name
mysql	index	Comment	string	Remarks about the index
mysql	index	Index_type	string	The index method used (BTREE, FULLTEXT, HASH, RTREE)
mysql	index	Key_name	string	The name of the index
mysql	index	Non_unique	long	Whether the index can contain duplicates
mysql	index	Null	string	Whether the column may contain NULL values
mysql	index	Packed	string	Whether the key is packed
mysql	index	Seq_in_index	long	The column sequence number in the index, starting with 1
mysql	index	Sub_part	string	The number of indexed characters if the column is only partly indexed
mysql	index	Table	string	The name of the table associated with the index
mysql	innodbstatus	innodb_bp_add_alloc	long	The total memory allocated for the additional buffer pool measured in bytes
mysql	innodbstatus	innodb_bp_created_per_sec	double	The number of buffer pool pages created per second
mysql	innodbstatus	innodb_bp_db_pages	long	The current number of buffer pool pages
mysql	innodbstatus	innodb_bp_free_buffers	long	The current number of free buffer pool pages
mysql	innodbstatus	innodb_bp_hit_rate	long	The buffer pool hit rate
mysql	innodbstatus	innodb_bp_modified_pages	long	The current number of pages modified

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodbstatus	innodb_bp_pages_created	long_counter	The total number of pages created
mysql	innodbstatus	innodb_bp_pages_read	long_counter	The total number of pages read
mysql	innodbstatus	innodb_bp_pages_written	long_counter	The total number of pages written
mysql	innodbstatus	innodb_bp_pending_pages	long	The number of pending page writes
mysql	innodbstatus	innodb_bp_pending_writes_flush_list	long	The number of pages to be flushed during checkpointing
mysql	innodbstatus	innodb_bp_pending_writes_lru	long	The number of old dirty pages to be written from the bottom of the LRU list
mysql	innodbstatus	innodb_bp_pending_writes_single_page	long	The number of pending independent page writes
mysql	innodbstatus	innodb_bp_reads_per_sec	double	The number of buffer pool reads per second
mysql	innodbstatus	innodb_bp_size	long	The total buffer pool size in bytes
mysql	innodbstatus	innodb_bp_total_alloc	long	The total memory allocated for the buffer pool
mysql	innodbstatus	innodb_bp_written_per_sec	double	The number of buffer pool pages written per second
mysql	innodbstatus	innodb_bytes_per_read	long	The number of bytes per read
mysql	innodbstatus	innodb_datetime	string	The date and time the SHOW ENGINE INNODB STATUS snapshot was taken
mysql	innodbstatus	innodb_hash_node_heap	long	Number of buffer pool pages reserved for the Adaptive Hash Index
mysql	innodbstatus	innodb_hash_searches_per_sec	double	The number of hash searches per second
mysql	innodbstatus	innodb_hash_table_size	long	The size of the hash table
mysql	innodbstatus	innodb_hash_used_cells	long	Number of buffer pool pages used for the Adaptive Hash Index
mysql	innodbstatus	innodb_ibuf_inserts	long_counter	The number of insert buffer inserts
mysql	innodbstatus	innodb_ibuf_merged_recs	long_counter	The number of insert buffer merged records
mysql	innodbstatus	innodb_ibuf_merges	long_counter	The number of insert buffer merges
mysql	innodbstatus	innodb_io_ibuf_logs	long	The number of pending log Ios
mysql	innodbstatus	innodb_io_ibuf_reads	long	The number of pending insert buffer reads
mysql	innodbstatus	innodb_io_ibuf_syncs	long	The number of pending synch operations
mysql	innodbstatus	innodb_io_os_file_fsyncs	long_counter	The number of OS fsyncs
mysql	innodbstatus	innodb_io_os_file_reads	long_counter	The number of OS file reads
mysql	innodbstatus	innodb_io_os_file_writes	long_counter	The number of OS file writes

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodbstatus	innodb_io_pending_flush_bp	long	The number of pending buffer pool flush operations
mysql	innodbstatus	innodb_io_pending_flush_log	long	The number of pending log flush operations
mysql	innodbstatus	innodb_io_pending_reads	long	The number of I/O pending reads
mysql	innodbstatus	innodb_io_pending_writes	long	The number of I/O pending writes
mysql	innodbstatus	innodb_io_syncs_per_sec	double	The number of fsyncs() per second
mysql	innodbstatus	innodb_log_checkpoint_file	long	The log file number the last checkpoint was performed on
mysql	innodbstatus	innodb_log_checkpoint_lsn	long	The log sequence number of the last checkpoint
mysql	innodbstatus	innodb_log_flushed_file	long	The log file number the last checkpoint was performed on
mysql	innodbstatus	innodb_log_flushed_lsn	long	The point up to which the log was last flushed
mysql	innodbstatus	innodb_log_io_per_sec	double	The number of log I/O's per second
mysql	innodbstatus	innodb_log_io_total	long_counter	The total number of log I/O's
mysql	innodbstatus	innodb_log_pending_checkpoint_writes	long	The log pending checkpoint writes
mysql	innodbstatus	innodb_log_pending_log_writes	long	The log pending log writes
mysql	innodbstatus	innodb_log_sequence_file	long	The log sequence file number
mysql	innodbstatus	innodb_log_sequence_lsn	long	The log sequence number
mysql	innodbstatus	innodb_non_hash_searches_per_sec	double	The number of nonadaptive hash index searches
mysql	innodbstatus	innodb_per_sec_avg	double	The number of seconds the averages for SHOW INNODB STATUS were calculated from
mysql	innodbstatus	innodb_reads_per_sec	double	The number of reads per second
mysql	innodbstatus	innodb_row_queries_inside	long	The number of queries executing inside InnoDB
mysql	innodbstatus	innodb_row_queries_queue	long	The number of queries in the queue, waiting to enter InnoDB
mysql	innodbstatus	innodb_row_state	string	The current state of the main InnoDB thread
mysql	innodbstatus	innodb_rows_deleted	long_counter	The number of rows deleted from InnoDB tables
mysql	innodbstatus	in-	double	The number of rows deleted per second

Namespace	Namespace Type	Attribute	Type	Description
		nodb_rows_deleted_per_sec		
mysql	innodbstatus	innodb_rows_inserted	long_counter	The number of rows inserted into InnoDB tables
mysql	innodbstatus	innodb_rows_inserted_per_sec	double	The number of rows inserted per second
mysql	innodbstatus	innodb_rows_read	long_counter	The number of rows read from InnoDB tables
mysql	innodbstatus	innodb_rows_read_per_sec	double	The number of rows read per second
mysql	innodbstatus	innodb_rows_updated	long_counter	The number of rows updated in InnoDB tables
mysql	innodbstatus	innodb_rows_updated_per_sec	double	The number of rows updated per second
mysql	innodbstatus	innodb_sem_mutex_os_waits	long_counter	The number of semaphore/mutex waits yielded to the OS
mysql	innodbstatus	innodb_sem_mutex_rounds	long_counter	The number of semaphore/mutex round for the internal sync array
mysql	innodbstatus	innodb_sem_mutex_spin_waits	long_counter	The number of semaphore/mutex spin waits for the internal sync array
mysql	innodbstatus	innodb_sem_os_reservation_count	long_counter	The number of times a mutex wait was added to the internal sync array
mysql	innodbstatus	innodb_sem_os_signal_count	long_counter	Performance.InnoDB
mysql	innodbstatus	innodb_sem_rw_excl_os_waits	long_counter	The number of exclusive (write) semaphore waits yielded to the OS
mysql	innodbstatus	innodb_sem_rw_excl_spins	long_counter	Performance.InnoDB
mysql	innodbstatus	innodb_sem_shared_os_waits	long_counter	The number of shared (read) semaphore waits yielded to the OS
mysql	innodbstatus	innodb_sem_shared_spins	long_counter	The number of shared (read) semaphore spin waits within the syn array
mysql	innodbstatus	innodb_trx_history_list_length	long	The transaction history list length
mysql	innodbstatus	innodb_trx_id_counter1	long	The transaction counter roll over variable
mysql	innodbstatus	innodb_trx_id_counter2	long	The main transaction count
mysql	innodbstatus	innodb_trx_purge_done_trx2	long	The transaction count that is already purged
mysql	innodbstatus	innodb_trx_total_lock_structs	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodbstatus	innodb_writes_per_sec	double	The number of InnoDB writes per second
mysql	masterlogs	File_size	long	The size of a binary log file
mysql	masterlogs	filecount	long	The number of binary log files
mysql	masterlogs	filesizesum	long	The total size of all the binary log files
mysql	masterlogs	Log_name	string	The name of the binary log file
mysql	MasterStatus	Binlog_Do_DB	string	
mysql	MasterStatus	Binlog_Ignore_DB	string	
mysql	MasterStatus	File	string	
mysql	MasterStatus	Position	long	
mysql	no_password	user	string	
mysql	no_root_password	no_password	long	
mysql	num_waiting_txns	num_waiting_txns	long	
mysql	prepared_count	prepared_count	long	
mysql	privileges_on_all_dbs	user_spec	string	
mysql	processlist	Command	string	The type of command the thread is executing
mysql	processlist	db	string	The default database, if one is selected
mysql	processlist	Host	string	The host name of the client issuing the statement
mysql	processlist	Id	long	The connection identifier
mysql	processlist	Info	string	The statement that the thread is executing
mysql	processlist	State	string	An action, event, or state that indicates what the thread is doing
mysql	processlist	Time	long	The time in seconds that the thread has been in its current state
mysql	processlist	User	string	The MySQL user who issued the statement
mysql	repl_slave_privileges	repl_slaves	long	
mysql	rolledback_count	rolledback_count	long	
mysql	root_account_exists	root_account	long	
mysql	root_remote_login	remote_login	long	
mysql	Server	agent.reachable	string	Whether the agent is reachable or not
mysql	Server	blackout	string	
mysql	Server	displayname	string	The display name of the server in the Dashboard
mysql	Server	registration-complete	string	
mysql	Server	repl.group	long	
mysql	Server	repl.groupName	string	
mysql	Server	server.connected	long	Whether the server is connected
mysql	Server	server.last_errno	string	
mysql	Server	server.last_error	string	The last MySQL server error message
mysql	Server	server.reachable	long	Whether the server is reachable
mysql	Server	server.version_numeric	long	The MySQL server version number
mysql	Server	Time	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	Server	transport	string	
mysql	Server	visible.displayname	string	
mysql	server_admin_privileges	user_spec	string	
mysql	SlaveStatus	Binlog_Do_DB	string	
mysql	SlaveStatus	Binlog_Ignore_DB	string	
mysql	SlaveStatus	Connect_Retry	long	
mysql	SlaveStatus	Exec_Master_Log_Pos	long	
mysql	SlaveStatus	File	string	
mysql	SlaveStatus	Last_Errno	long	
mysql	SlaveStatus	Last_Error	string	
mysql	SlaveStatus	Master_Host	string	
mysql	SlaveStatus	Master_ip	string	
mysql	SlaveStatus	Master_Log_File	string	
mysql	SlaveStatus	Master_Port	long	
mysql	SlaveStatus	Master_SSL_Allowed	string	
mysql	SlaveStatus	Master_SSL_CA_File	string	
mysql	SlaveStatus	Master_SSL_CA_Path	string	
mysql	SlaveStatus	Master_SSL_Cert	string	
mysql	SlaveStatus	Master_SSL_Cipher	string	
mysql	SlaveStatus	Master_SSL_Key	string	
mysql	SlaveStatus	Master_User	string	
mysql	SlaveStatus	Master_uuid	string	
mysql	SlaveStatus	Position	long	
mysql	SlaveStatus	Read_Master_Log_Pos	long	
mysql	SlaveStatus	Relay_Log_File	string	
mysql	SlaveStatus	Relay_Log_Pos	long	
mysql	SlaveStatus	Relay_Log_Space	long	
mysql	SlaveStatus	Relay_Master_Log_File	string	
mysql	SlaveStatus	Replicate_Do_DB	string	
mysql	SlaveStatus	Replicate_Do_Table	string	
mysql	SlaveStatus	Replicate_Ignore_DB	string	
mysql	SlaveStatus	Replicate_Ignore_Table	string	
mysql	SlaveStatus	Replicate_Wild_Do_Table	string	
mysql	SlaveStatus	Replicate_Wild_Ignore_Table	string	
mysql	SlaveStatus	Seconds_Behind_Master	long	
mysql	SlaveStatus	Skip_Counter	long	
mysql	SlaveStatus	Slave_IO_Running	string	
mysql	SlaveStatus	Slave_IO_State	string	
mysql	SlaveStatus	Slave_SQL_Running	string	
mysql	SlaveStatus	Until_Condition	string	
mysql	SlaveStatus	Until_Log_File	string	
mysql	SlaveStatus	Until_Log_Pos	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	sp_with_select_star	routine	string	
mysql	Statement	bytes	long	
mysql	Statement	comment	string	
mysql	Statement	connection_id	long	
mysql	Statement	exec_time	long	
mysql	Statement	host_from	string	
mysql	Statement	host_to	string	
mysql	Statement	max_bytes	long	
mysql	Statement	rows	long	
mysql	Statement	text	string	
mysql	Statement	user	string	
mysql	StatementAnalysisSupport	capture_examples	string	
mysql	StatementAnalysisSupport	capture_explain	string	
mysql	StatementAnalysisSupport	enabled	string	
mysql	StatementAnalysisSupport	frequency	string	
mysql	StatementSummary	bytes	long	
mysql	StatementSummary	bytes_rel	long	
mysql	StatementSummary	count	long	
mysql	StatementSummary	count_rel	long	
mysql	StatementSummary	database	string	
mysql	StatementSummary	exec_time	long	
mysql	StatementSummary	exec_time_rel	long	
mysql	StatementSummary	max_bytes	long	
mysql	StatementSummary	max_exec_time	long	
mysql	StatementSummary	max_rows	long	
mysql	StatementSummary	min_bytes	long	
mysql	StatementSummary	min_exec_time	long	
mysql	StatementSummary	min_rows	long	
mysql	StatementSummary	query_type	string	
mysql	StatementSummary	rows	long	
mysql	StatementSummary	rows_rel	long	
mysql	StatementSummary	text	string	
mysql	StatementSummary	text_hash	string	
mysql	status	Aborted_clients	long_counter	Networking.Overview
mysql	status	Aborted_connects	long_counter	The number of failed attempts to connect to the MySQL server
mysql	status	Binlog_cache_disk_use	long	General.Logging
mysql	status	Binlog_cache_use	long	The number of transactions that used the temporary binary log cache
mysql	status	Bytes_received	long_counter	The number of bytes received from all clients
mysql	status	Bytes_sent	long_counter	The number of bytes sent to all clients
mysql	status	Com_admin_commands	long_counter	Count of admin commands

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_alter_db	long_counter	Count of ALTER DATABASE statements
mysql	status	Com_alter_db_upgrade	long_counter	
mysql	status	Com_alter_event	long_counter	
mysql	status	Com_alter_function	long_counter	
mysql	status	Com_alter_procedure	long_counter	
mysql	status	Com_alter_server	long_counter	
mysql	status	Com_alter_table	long_counter	Count of ALTER TABLE statements
mysql	status	Com_alter_tablespace	long_counter	
mysql	status	Com_analyze	long_counter	Count of ANALYZE statements
mysql	status	Com_assign_to_keycache	long_counter	
mysql	status	Com_backup_table	long_counter	Count of BACKUP TABLE statements
mysql	status	Com_begin	long_counter	Count of BEGIN statements
mysql	status	Com_binlog	long_counter	
mysql	status	Com_call_procedure	long_counter	Number of calls to stored procedures
mysql	status	Com_change_db	long_counter	Count of CHANGE DATABASE statements
mysql	status	Com_change_master	long_counter	Count of CHANGE MASTER statements
mysql	status	Com_check	long_counter	Count of CHECK statements
mysql	status	Com_checksum	long_counter	Count of CHECKSUM statements
mysql	status	Com_commit	long_counter	Count of COMMIT statements
mysql	status	Com_create_db	long_counter	Count of CREATE DATABASE statements
mysql	status	Com_create_event	long_counter	
mysql	status	Com_create_function	long_counter	Count of CREATE FUNCTION statements
mysql	status	Com_create_index	long_counter	Count of CREATE INDEX statements
mysql	status	Com_create_procedure	long_counter	
mysql	status	Com_create_server	long_counter	
mysql	status	Com_create_table	long_counter	Count of CREATE TABLE statements
mysql	status	Com_create_trigger	long_counter	
mysql	status	Com_create_udf	long_counter	
mysql	status	Com_create_user	long_counter	Count of CREATE USER statements
mysql	status	Com_create_view	long_counter	
mysql	status	Com_dealloc_sql	long_counter	Count of DEALLOCATE PREPARE statements
mysql	status	Com_delete	long_counter	Count of DELETE statements
mysql	status	Com_delete_multi	long_counter	Count of multiple table DELETE statements
mysql	status	Com_do	long_counter	Count of DO statements
mysql	status	Com_drop_db	long_counter	Count of DROP DATABASE statements
mysql	status	Com_drop_event	long_counter	
mysql	status	Com_drop_function	long_counter	Count of DROP FUNCTION statements
mysql	status	Com_drop_index	long_counter	Count of DROP INDEX statements

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_drop_procedure	long_counter	
mysql	status	Com_drop_server	long_counter	
mysql	status	Com_drop_table	long_counter	Count of DROP TABLE statements
mysql	status	Com_drop_trigger	long_counter	
mysql	status	Com_drop_user	long_counter	Count of DROP USER statements
mysql	status	Com_drop_view	long_counter	
mysql	status	Com_empty_query	long_counter	
mysql	status	Com_execute_sql	long_counter	Count of EXECUTE statements
mysql	status	Com_flush	long_counter	Count of FLUSH statements
mysql	status	Com_grant	long_counter	Count of GRANT statements
mysql	status	Com_ha_close	long_counter	Count of HANDLER CLOSE statements
mysql	status	Com_ha_open	long_counter	Count of HANDLER OPEN statements
mysql	status	Com_ha_read	long_counter	Count of HANDLER READ statements
mysql	status	Com_help	long_counter	Count of HELP statements
mysql	status	Com_insert	long_counter	Count of INSERT statements
mysql	status	Com_insert_select	long_counter	Count of INSERT SELECT statements
mysql	status	Com_install_plugin	long_counter	
mysql	status	Com_kill	long_counter	Count of KILL statements
mysql	status	Com_load	long_counter	Count of LOAD statements
mysql	status	Com_load_master_data	long_counter	Count of LOAD MASTER DATA statements
mysql	status	Com_load_master_table	long_counter	Count of LOAD MASTER TABLE statements
mysql	status	Com_lock_tables	long_counter	Count of LOCK TABLES statements
mysql	status	Com_optimize	long_counter	Count of OPTIMIZE statements
mysql	status	Com_preload_keys	long_counter	Count of PRELOAD KEYS statements
mysql	status	Com_prepare_sql	long_counter	Count of PREPARE statements
mysql	status	Com_purge	long_counter	Count of PURGE statements
mysql	status	Com_purge_before_date	long_counter	Count of PURGE BEFORE DATE statements
mysql	status	Com_release_savepoint	long_counter	
mysql	status	Com_rename_table	long_counter	Count of RENAME TABLE statements
mysql	status	Com_rename_user	long_counter	
mysql	status	Com_repair	long_counter	Count of REPAIR statements
mysql	status	Com_replace	long_counter	Count of REPLACE statements
mysql	status	Com_replace_select	long_counter	Count of REPLACE SELECT statements
mysql	status	Com_reset	long_counter	Count of RESET statements
mysql	status	Com_restore_table	long_counter	Count of RESTORE TABLE statements
mysql	status	Com_revoke	long_counter	Count of REVOKE statements
mysql	status	Com_revoke_all	long_counter	Count of REVOKE ALL statements
mysql	status	Com_rollback	long_counter	Count of ROLLBACK statements
mysql	status	Com_rollback_to_savepoint	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_savepoint	long_counter	Count of SAVEPOINT statements
mysql	status	Com_select	long_counter	Count of SELECT statements
mysql	status	Com_set_option	long_counter	Count of SET OPTION statements
mysql	status	Com_show_authors	long_counter	
mysql	status	Com_show_binlog_events	long_counter	Count of SHOW BINLOG EVENTS statements
mysql	status	Com_show_binlogs	long_counter	Count of SHOW BINLOGS statements
mysql	status	Com_show_charsets	long_counter	Count of SHOW CHARSETS statements
mysql	status	Com_show_collations	long_counter	Count of SHOW COLLATIONS statements
mysql	status	Com_show_column_types	long_counter	Count of SHOW COLUMN TYPES statements
mysql	status	Com_show_contributors	long_counter	
mysql	status	Com_show_create_db	long_counter	Count of SHOW CREATE DATABASE statements
mysql	status	Com_show_create_event	long_counter	
mysql	status	Com_show_create_func	long_counter	
mysql	status	Com_show_create_proc	long_counter	
mysql	status	Com_show_create_table	long_counter	Count of SHOW CREATE TABLE statements
mysql	status	Com_show_create_trigger	long_counter	
mysql	status	Com_show_databases	long_counter	Count of SHOW DATABASES statements
mysql	status	Com_show_engine_logs	long_counter	
mysql	status	Com_show_engine_mutex	long_counter	
mysql	status	Com_show_engine_status	long_counter	
mysql	status	Com_show_errors	long_counter	Count of SHOW ERRORS statements
mysql	status	Com_show_events	long_counter	
mysql	status	Com_show_fields	long_counter	Count of SHOW FIELDS statements
mysql	status	Com_show_function_status	long_counter	
mysql	status	Com_show_grants	long_counter	Count of SHOW GRANTS statements
mysql	status	Com_show_innodb_status	long_counter	Count of SHOW INNODB STATUS statements
mysql	status	Com_show_keys	long_counter	Count of SHOW KEYS statements
mysql	status	Com_show_logs	long_counter	Count of SHOW LOGS statements
mysql	status	Com_show_master_status	long_counter	Count of SHOW MASTER STATUS statements
mysql	status	Com_show_ndb_status	long_counter	Count of SHOW NDB STATUS statements
mysql	status	Com_show_new_master	long_counter	Count of SHOW NEW MASTER statements
mysql	status	Com_show_open_tables	long_counter	Count of SHOW OPEN TABLES statements
mysql	status	Com_show_plugins	long_counter	
mysql	status	Com_show_privileges	long_counter	Count of SHOW PRIVILEGES state-

Namespace	Namespace Type	Attribute	Type	Description
				ments
mysql	status	Com_show_procedure_status	long_counter	
mysql	status	Com_show_processlist	long_counter	Count of SHOW PROCESSLIST statements
mysql	status	Com_show_profile	long_counter	
mysql	status	Com_show_profiles	long_counter	
mysql	status	Com_show_slave_hosts	long_counter	Count of SHOW SLAVE HOSTS statements
mysql	status	Com_show_slave_status	long_counter	Count of SHOW SLAVE STATUS statements
mysql	status	Com_show_status	long_counter	Count of SHOW STATUS statements
mysql	status	Com_show_storage_engines	long_counter	Count of SHOW STORAGE ENGINES statements
mysql	status	Com_show_table_status	long_counter	
mysql	status	Com_show_tables	long_counter	Count of SHOW TABLES statements
mysql	status	Com_show_triggers	long_counter	Count of SHOW TRIGGERS statements
mysql	status	Com_show_variables	long_counter	Count of SHOW VARIABLES statements
mysql	status	Com_show_warnings	long_counter	Count of SHOW WARNINGS statements
mysql	status	Com_slave_start	long_counter	Count of SLAVE START statements
mysql	status	Com_slave_stop	long_counter	Count of SLAVE STOP statements
mysql	status	Com_stmt_close	long_counter	Count of STATEMENT CLOSE statements
mysql	status	Com_stmt_execute	long_counter	Count of STATEMENT EXECUTE statements
mysql	status	Com_stmt_fetch	long_counter	Count of STATEMENT FETCH statements
mysql	status	Com_stmt_prepare	long_counter	Count of STATEMENT PREPARE statements
mysql	status	Com_stmt_reprepare	long_counter	
mysql	status	Com_stmt_reset	long_counter	Count of STATEMENT RESET statements
mysql	status	Com_stmt_send_long_data	long_counter	Count of STATEMENT SEND LONG DATA statements
mysql	status	Com_truncate	long_counter	Count of TRUNCATE statements
mysql	status	Com_uninstall_plugin	long_counter	
mysql	status	Com_unlock_tables	long_counter	Count of UNLOCK TABLES statements
mysql	status	Com_update	long_counter	Count of UPDATE statements
mysql	status	Com_update_multi	long_counter	Count of multiple UPDATE statements
mysql	status	Com_xa_commit	long_counter	Count of XA COMMIT statements
mysql	status	Com_xa_end	long_counter	Count of XA END statements
mysql	status	Com_xa_prepare	long_counter	Count of XA PREPARE statements
mysql	status	Com_xa_recover	long_counter	Count of XA RECOVER statements
mysql	status	Com_xa_rollback	long_counter	Count of XA ROLLBACK statements
mysql	status	Com_xa_start	long_counter	Count of XA START statements

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Compression	string	General.Features
mysql	status	Connections	long_counter	The number of connection attempts
mysql	status	Created_tmp_disk_tables	long_counter	General.Temporary
mysql	status	Created_tmp_files	long_counter	How many temporary files mysqld has created
mysql	status	Created_tmp_tables	long_counter	How many temporary tables mysqld has created
mysql	status	Delayed_errors	long_counter	Performance.Delayed
mysql	status	Delayed_insert_threads	long	The number of INSERT DELAYED thread handlers in use
mysql	status	Delayed_writes	long_counter	The number of INSERT DELAYED rows written
mysql	status	displayname	string	The display name of the server in the Dashboard
mysql	status	Flush_commands	long_counter	The number of FLUSH statements executed
mysql	status	groupname	string	The name of the group to which the server belongs
mysql	status	Handler_commit	long_counter	The number of internal COMMIT statements
mysql	status	Handler_delete	long_counter	The number of times that rows have been deleted from tables
mysql	status	Handler_discover	long_counter	Indicates the number of times that tables have been discovered
mysql	status	Handler_prepare	long_counter	A counter for the prepare phase of two-phase commit operations
mysql	status	Handler_read_first	long_counter	The number of times the first entry was read from an index
mysql	status	Handler_read_key	long_counter	The number of requests to read a row based on a key
mysql	status	Handler_read_next	long_counter	The number of requests to read the next row in key order
mysql	status	Handler_read_prev	long_counter	The number of requests to read the previous row in key order
mysql	status	Handler_read_rnd	long_counter	The number of requests to read a row based on a fixed position
mysql	status	Handler_read_rnd_next	long_counter	The number of requests to read the next row in the data file
mysql	status	Handler_rollback	long_counter	Miscellaneous.Handler
mysql	status	Handler_savepoint	long_counter	The number of requests for a storage engine to place a savepoint
mysql	status	Handler_savepoint_rollback	long_counter	The number of requests for a storage engine to roll back to a savepoint
mysql	status	Handler_update	long_counter	The number of requests to update a row in a table
mysql	status	Handler_write	long_counter	The number of requests to insert a row in a table
mysql	status	Innodb_buffer_pool_pages_data	long	The number of pages containing data (dirty or clean)

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Innodb_buffer_pool_pages_dirty	long	The number of pages currently dirty
mysql	status	Innodb_buffer_pool_pages_flushed	long_counter	The number of buffer pool page-flush requests
mysql	status	Innodb_buffer_pool_pages_free	long	The number of buffer pool pages free
mysql	status	Innodb_buffer_pool_pages_latched	long	The number of latched pages in InnoDB buffer pool
mysql	status	Innodb_buffer_pool_pages_misc	long	Performance.InnoDB
mysql	status	Innodb_buffer_pool_pages_total	long	The total size of the buffer pool, in pages
mysql	status	Innodb_buffer_pool_read_ahead_rnd	long_counter	The number of random read-aheads initiated by InnoDB
mysql	status	Innodb_buffer_pool_read_ahead_seq	long_counter	The number of sequential read-aheads initiated by InnoDB
mysql	status	Innodb_buffer_pool_read_requests	long_counter	The number of logical read requests InnoDB has done
mysql	status	Innodb_buffer_pool_reads	long_counter	Performance.InnoDB
mysql	status	Innodb_buffer_pool_wait_free	long_counter	Counts number of waits for pages to be flushed
mysql	status	Innodb_buffer_pool_write_requests	long_counter	The number of writes done to the buffer pool
mysql	status	Innodb_data_fsyncs	long_counter	The number of fsync() operations so far
mysql	status	Innodb_data_pending_fsyncs	long	The current number of pending fsync() operations
mysql	status	Innodb_data_pending_reads	long	The current number of pending reads
mysql	status	Innodb_data_pending_writes	long	The number of pending writes
mysql	status	Innodb_data_read	long_counter	The amount of data read so far, in bytes
mysql	status	Innodb_data_reads	long_counter	The total number of data reads
mysql	status	Innodb_data_writes	long_counter	The total number of data writes
mysql	status	Innodb_data_written	long_counter	The amount of data written in bytes
mysql	status	Innodb_dblwr_pages_written	long_counter	The number of doublewrite pages that have been written
mysql	status	Innodb_dblwr_writes	long_counter	The number of doublewrite operations

Namespace	Namespace Type	Attribute	Type	Description
				that have been performed
mysql	status	Innodb_log_waits	long_counter	Performance.InnoDB
mysql	status	Innodb_log_write_requests	long_counter	The number of log write requests
mysql	status	Innodb_log_writes	long_counter	The number of physical writes to the log
mysql	status	Innodb_os_log_fsyncs	long_counter	The number of fsync() writes done to the log file
mysql	status	Innodb_os_log_pending_fsyncs	long	The number of pending log file fsync() operations
mysql	status	Innodb_os_log_pending_writes	long	The number of pending log file writes
mysql	status	Innodb_os_log_written	long_counter	The number of bytes written to the log file
mysql	status	Innodb_page_size	long	The compiled-in InnoDB page size
mysql	status	Innodb_pages_created	long_counter	The number of pages created
mysql	status	Innodb_pages_read	long_counter	The number of pages read
mysql	status	Innodb_pages_written	long_counter	The number of pages written
mysql	status	Innodb_row_lock_current_waits	long	The number of row locks currently being waited for
mysql	status	Innodb_row_lock_time	long_counter	The total time spent in acquiring row locks, in milliseconds
mysql	status	Innodb_row_lock_time_avg	long	The average time to acquire a row lock, in milliseconds
mysql	status	Innodb_row_lock_time_max	long	The maximum time to acquire a row lock, in milliseconds
mysql	status	Innodb_row_lock_waits	long_counter	The number of times a row lock had to be waited for
mysql	status	Innodb_rows_deleted	long_counter	The number of rows deleted from InnoDB tables
mysql	status	Innodb_rows_inserted	long_counter	The number of rows inserted into InnoDB tables
mysql	status	Innodb_rows_read	long_counter	The number of rows read from InnoDB tables
mysql	status	Innodb_rows_updated	long_counter	The number of rows updated in InnoDB tables
mysql	status	Key_blocks_not_flushed	long	Performance.Keys
mysql	status	Key_blocks_unused	long	The number of unused blocks in the key cache
mysql	status	Key_blocks_used	long	The number of used blocks in the key cache
mysql	status	Key_read_requests	long_counter	The number of requests to read a key block from the cache
mysql	status	Key_reads	long_counter	The number of physical reads of a key block from disk

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Key_write_requests	long_counter	The number of requests to write a key block to the cache
mysql	status	Key_writes	long_counter	The number of physical writes of a key block to disk
mysql	status	Last_query_cost	long	Performance.Cost
mysql	status	Max_used_connections	long	Networking.Overview
mysql	status	Not_flushed_delayed_rows	long	The number of rows waiting to be written in INSERT DELAY queues
mysql	status	Open_files	long	The number of files that are open
mysql	status	Open_streams	long	The number of streams that are open (used mainly for logging)
mysql	status	Open_table_definitions	long	
mysql	status	Open_tables	long	The number of tables that are open
mysql	status	Opened_files	long_counter	
mysql	status	Opened_table_definitions	long_counter	
mysql	status	Opened_tables	long_counter	The number of tables that have been opened
mysql	status	Prepared_stmt_count	long	The current number of prepared statements
mysql	status	Qcache_free_blocks	long	The number of free memory blocks in the query cache
mysql	status	Qcache_free_memory	long	The amount of free memory for the query cache
mysql	status	Qcache_hits	long_counter	The number of query cache hits
mysql	status	Qcache_inserts	long_counter	The number of query cache inserts
mysql	status	Qcache_lowmem_prunes	long_counter	Performance.Query Cache
mysql	status	Qcache_not_cached	long_counter	Performance.Query Cache
mysql	status	Qcache_queries_in_cache	long	The number of queries registered in the query cache
mysql	status	Qcache_total_blocks	long	The total number of blocks in the query cache
mysql	status	Questions	long_counter	The number of statements that clients have sent to the server
mysql	status	Rpl_status	long	The status of fail-safe replication (not yet implemented)
mysql	status	Select_full_join	long_counter	Performance.Selects
mysql	status	Select_full_range_join	long_counter	The number of joins that used a range search on a reference table
mysql	status	Select_range	long_counter	The number of joins that used ranges on the first table
mysql	status	Select_range_check	long_counter	The number of joins without keys that check for key usage after each row
mysql	status	Select_scan	long_counter	The number of joins that did a full scan of the first table
mysql	status	Slave_open_temp_tables	long	Networking.Replication
mysql	status	Slave_retried_transactions	long_counter	Networking.Replication
mysql	status	Slave_running	string	This is ON if this server is a slave that

Namespace	Namespace Type	Attribute	Type	Description
				is connected to a master
mysql	status	`Slow_launch_threads`	`long_counter`	Performance.Slow Items
mysql	status	`Slow_queries`	`long_counter`	The number of queries that have taken more than long_query_time seconds
mysql	status	`Sort_merge_passes`	`long_counter`	The number of merge passes that the sort algorithm has had to do
mysql	status	`Sort_range`	`long_counter`	The number of sorts that were done using ranges
mysql	status	`Sort_rows`	`long_counter`	The number of sorted rows
mysql	status	`Sort_scan`	`long_counter`	The number of sorts that were done by scanning the table
mysql	status	`Ssl_accept_renegotiates`	`long_counter`	The number of renegotiates needed to establish the connection
mysql	status	`Ssl_accepts`	`long_counter`	The number of attempted SSL connections
mysql	status	`Ssl_callback_cache_hits`	`long_counter`	The number of callback cache hits
mysql	status	`Ssl_cipher`	`string`	SSL cipher to use (implies --ssl)
mysql	status	`Ssl_cipher_list`	`string`	The list of SSL cipher strings
mysql	status	`Ssl_client_connects`	`long_counter`	The number of attempted connections to an SSL enabled master
mysql	status	`Ssl_connect_renegotiates`	`long_counter`	Number of renegotiates needed to connect to an SSL-enabled master
mysql	status	`Ssl_ctx_verify_depth`	`long`	The SSL context verification depth
mysql	status	`Ssl_ctx_verify_mode`	`long`	The SSL certificate verification mode used by the server
mysql	status	`Ssl_default_timeout`	`long`	The SSL default timeout
mysql	status	`Ssl_finished_accepts`	`long_counter`	The number of successful SSL connections to the server
mysql	status	`Ssl_finished_connects`	`long_counter`	The number of successful SSL slave connections to the server
mysql	status	`Ssl_session_cache_hits`	`long_counter`	The number of SSL session cache hits
mysql	status	`Ssl_session_cache_misses`	`long_counter`	The number of SSL session cache misses
mysql	status	`Ssl_session_cache_mode`	`string`	The SSL session cache mode
mysql	status	`Ssl_session_cache_overflows`	`long_counter`	The number of SSL cache overflows
mysql	status	`Ssl_session_cache_size`	`long`	The size of the SSL session cache
mysql	status	`Ssl_session_cache_timeouts`	`long_counter`	The number of session cache timeouts
mysql	status	`Ssl_sessions_reused`	`long`	The number of SSL sessions from the cache that were reused
mysql	status	`Ssl_used_session_cache_entries`	`long`	The number of SSL session cache entries used
mysql	status	`Ssl_verify_depth`	`long`	Depth of verification for replication SSL connections
mysql	status	`Ssl_verify_mode`	`long`	Mode of veritifaction for replication SSL connections

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Ssl_version	string	The SSL version number
mysql	status	Table_locks_immediate	long_counter	The number of times that a table lock was acquired immediately
mysql	status	Table_locks_waited	long_counter	Performance.Locks
mysql	status	Tc_log_max_pages_used	long	General.Logging
mysql	status	Tc_log_page_size	long	General.Logging
mysql	status	Tc_log_page_waits	long	General.Logging
mysql	status	Threads_cached	long	The number of threads in the thread cache
mysql	status	Threads_connected	long	The number of currently open connections
mysql	status	Threads_created	long_counter	The number of threads created to handle connections
mysql	status	Threads_running	long	The number of threads that are not sleeping
mysql	status	Uptime	long	The number of seconds the server has been up
mysql	strong_privileges	user	string	
mysql	table	num_rows	long	MySQL Network.Table Data
mysql	table	table_auto_increment	long	The next AUTO_INCREMENT value
mysql	table	table_avg_row_length	long	The average row length
mysql	table	table_collation	string	The table's character set and collation
mysql	table	table_comment	string	The table comment
mysql	table	table_create_time	string	When the data file was created
mysql	table	table_data_free	long	The number of allocated but unused bytes
mysql	table	table_data_length	long	The length of the data file
mysql	table	table_engine	string	The storage engine used by a table
mysql	table	table_index_length	long	The length of the index file
mysql	table	table_max_data_length	long	The maximum length of the data file
mysql	table	table_name	string	The name of a table
mysql	table	table_row_format	string	The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact)
mysql	table	table_version	long	The version number of the table's .frm file
mysql	tablestatus	Auto_increment	long	The next AUTO_INCREMENT value
mysql	tablestatus	Avg_row_length	long	The average row length
mysql	tablestatus	Check_time	string	When the table was last checked
mysql	tablestatus	Checksum	string	The live checksum value (if any)
mysql	tablestatus	Collation	string	The table's character set
mysql	tablestatus	Comment	string	The comment used when creating the table
mysql	tablestatus	Create_options	string	Extra options used with CREATE TABLE
mysql	tablestatus	Create_time	string	When the table was created
mysql	tablestatus	Data_free	long	The number of allocated but unused bytes

Namespace	Namespace Type	Attribute	Type	Description
mysql	tablestatus	Data_length	long	The length of the data file
mysql	tablestatus	Engine	string	The storage engine for the table
mysql	tablestatus	Index_length	long	The length of the index file
mysql	tablestatus	Max_data_length	long	The maximum length of the data file
mysql	tablestatus	Name	string	The table name
mysql	tablestatus	Row_format	string	The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact).
mysql	tablestatus	Rows	long	The number of rows in the table
mysql	tablestatus	Update_time	string	When the data file was last updated
mysql	tablestatus	Version	long	The version number of the table's .frm file.
mysql	test_database	Database (test)	string	
mysql	transactions_in_serial_log	transactions_in_serial_log	long	
mysql	trigger_with_select_star	trigger_defn	string	
mysql	user_on_missing_db	db_name	string	
mysql	user_on_missing_db	user	string	
mysql	variables	auto_increment_increment	long	Auto-increment columns are incremented by this value
mysql	variables	auto_increment_offset	long	Offset added to Auto-increment columns
mysql	variables	autocommit	string	
mysql	variables	automatic_sp_privileges	string	Creating and dropping stored procedures alters ACLs
mysql	variables	back_log	long	The number of outstanding connection requests MySQL can have
mysql	variables	basedir	string	General.Directories / Files
mysql	variables	big_tables	string	
mysql	variables	binlog_cache_size	long	Memory.Caches
mysql	variables	binlog_format	string	
mysql	variables	bulk_insert_buffer_size	long	Size of tree cache used in bulk insert optimization
mysql	variables	character_set_client	string	Current client character set
mysql	variables	character_set_connection	string	Current connection character set
mysql	variables	character_set_database	string	The character set used by the default database
mysql	variables	character_set_filesystem	string	Set the file system character set
mysql	variables	character_set_results	string	Current result character set
mysql	variables	character_set_server	string	SQL.Charsets
mysql	variables	character_set_system	string	The character set used by the server for storing identifiers
mysql	variables	character_sets_dir	string	Directory where character sets are
mysql	variables	collation_connection	string	The collation of the connection
mysql	variables	collation_database	string	The collation used by the default database
mysql	variables	collation_server	string	Set the default collation

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	completion_type	long	Default completion type
mysql	variables	concurrent_insert	string	Use concurrent insert with MyISAM
mysql	variables	connect_timeout	long	Connections.Overview
mysql	variables	datadir	string	Path to the database root
mysql	variables	date_format	string	The DATE format (For future)
mysql	variables	datetime_format	string	The DATETIME/TIMESTAMP format (for future)
mysql	variables	default_week_format	long	The default week format used by WEEK() functions
mysql	variables	delay_key_write	string	Type of DELAY_KEY_WRITE
mysql	variables	delayed_insert_limit	long	General.Performance
mysql	variables	delayed_insert_timeout	long	General.Performance
mysql	variables	delayed_queue_size	long	General.Performance
mysql	variables	div_precision_increment	long	SQL.Formats
mysql	variables	engine_condition_pushdown	string	Push supported query conditions to the storage engine
mysql	variables	error_count	long	
mysql	variables	event_scheduler	string	
mysql	variables	expire_logs_days	long	General.Miscellaneous
mysql	variables	flush	string	Flush tables to disk between SQL commands
mysql	variables	flush_time	long	A dedicated thread is created to flush all tables at the given interval
mysql	variables	foreign_key_checks	string	
mysql	variables	ft_boolean_syntax	string	List of operators for MATCH ... AGAINST (... IN BOOLEAN MODE)
mysql	variables	ft_max_word_len	long	SQL.Full Text Search
mysql	variables	ft_min_word_len	long	SQL.Full Text Search
mysql	variables	ft_query_expansion_limit	long	Number of best matches to use for query expansion
mysql	variables	ft_stopword_file	string	Use stopwords from this file instead of built-in list
mysql	variables	general_log	string	
mysql	variables	general_log_file	string	
mysql	variables	group_concat_max_len	long	The maximum length of the result of function group_concat
mysql	variables	have_archive	string	Whether mysqld supports archive tables
mysql	variables	have_bdb	string	Is Berkeley DB supported
mysql	variables	have_blackhole_engine	string	Whether mysqld supports BLACKHOLE tables
mysql	variables	have_community_features	string	
mysql	variables	have_compress	string	Availability of the zlib compression library
mysql	variables	have_crypt	string	Availability of the crypt() system call

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	have_csv	string	Whether mysqld supports csv tables
mysql	variables	have_dynamic_loading	string	Whether mysqld supports dynamic loading of plugins
mysql	variables	have_example_engine	string	Whether mysqld supports EXAMPLE tables
mysql	variables	have_federated_engine	string	Whether mysqld supports FEDERATED tables
mysql	variables	have_geometry	string	Whether mysqld supports spatial data types
mysql	variables	have_innodb	string	Whether mysqld supports InnoDB tables
mysql	variables	have_isam	string	Whether mysqld supports isam tables
mysql	variables	have_merge_engine	string	Whether mysqld supports merge tables
mysql	variables	have_ndbcluster	string	Whether mysqld supports NDB Cluster tables
mysql	variables	have_openssl	string	Whether mysqld supports ssl connections
mysql	variables	have_partitioning	string	
mysql	variables	have_query_cache	string	Whether mysqld supports query cache
mysql	variables	have_raid	string	Whether mysqld supports the RAID option
mysql	variables	have_rtree_keys	string	General.Features
mysql	variables	have_ssl	string	Whether the server supports an SSL connection
mysql	variables	have_symlink	string	Is symbolic link support enabled
mysql	variables	hostname	string	The name of the server host
mysql	variables	identity	long	
mysql	variables	init_connect	string	Command(s) that are executed for each new connection
mysql	variables	init_file	string	Read SQL commands from this file at startup
mysql	variables	init_slave	string	Command(s) that are executed when a slave connects to a master
mysql	variables	innodb_adaptive_hash_index	string	
mysql	variables	innodb_additional_mem_pool_size	long	Memory.Buffers
mysql	variables	innodb_autoextend_increment	long	Data file autoextend increment in megabytes
mysql	variables	innodb_autoinc_lock_mode	long	
mysql	variables	innodb_buffer_pool_awe_mem_mb	long	Memory.Buffers
mysql	variables	innodb_buffer_pool_size	long	Memory.Buffers
mysql	variables	innodb_checksums	string	Enable InnoDB checksums validation

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	innodb_commit_concurrency	long	Helps in performance tuning in heavily concurrent environments
mysql	variables	innodb_concurrency_tickets	long	Table Types.InnoDB
mysql	variables	innodb_data_file_path	string	Path to individual files and their sizes
mysql	variables	innodb_data_home_dir	string	The common part for InnoDB table spaces
mysql	variables	innodb_doublewrite	string	Enable InnoDB doublewrite buffer
mysql	variables	innodb_fast_shutdown	long	Speeds up the shutdown process of the InnoDB storage engine
mysql	variables	innodb_file_io_threads	long	Number of file I/O threads in InnoDB
mysql	variables	innodb_file_per_table	string	Stores each InnoDB table to an .ibd file in the database dir
mysql	variables	innodb_flush_log_at_trx_commit	long	Table Types.InnoDB
mysql	variables	innodb_flush_method	string	With which method to flush data
mysql	variables	innodb_force_recovery	long	Table Types.InnoDB
mysql	variables	innodb_lock_wait_timeout	long	Table Types.InnoDB
mysql	variables	innodb_locks_unsafe_for_binlog	string	Table Types.InnoDB
mysql	variables	innodb_log_arch_dir	string	Where full logs should be archived
mysql	variables	innodb_log_archive	string	Set to 1 if you want to have logs archived
mysql	variables	innodb_log_buffer_size	long	General.Logs
mysql	variables	innodb_log_file_size	long	Size of each log file in a log group
mysql	variables	innodb_log_files_in_group	long	General.Logs
mysql	variables	innodb_log_group_home_dir	string	Path to InnoDB log files
mysql	variables	innodb_max_dirty_pages_pct	long	Percentage of dirty pages allowed in bufferpool
mysql	variables	innodb_max_purge_lag	long	Desired maximum length of the purge queue (0 = no limit)
mysql	variables	innodb_mirrored_log_groups	long	Number of identical copies of log groups we keep for the database
mysql	variables	innodb_open_files	long	The maximum number of files that InnoDB keeps open at the same time
mysql	variables	innodb_rollback_on_timeout	string	Unknown.Unknown
mysql	variables	innodb_support_xa	string	Enable InnoDB support for the XA

Namespace	Namespace Type	Attribute	Type	Description
				two-phase commit
mysql	variables	innodb_sync_spin_loops	long	Count of spin-loop rounds in InnoDB mutexes
mysql	variables	innodb_table_locks	string	Enable InnoDB locking in LOCK TABLES
mysql	variables	innodb_thread_concurrency	long	Table Types.InnoDB
mysql	variables	innodb_thread_sleep_delay	long	Table Types.InnoDB
mysql	variables	insert_id	long	
mysql	variables	interactive_timeout	long	Connections.Overview
mysql	variables	join_buffer_size	long	The size of the buffer that is used for full joins
mysql	variables	keep_files_on_create	string	
mysql	variables	key_buffer_size	long	The size of the buffer used for index blocks for MyISAM tables
mysql	variables	key_cache_age_threshold	long	Memory.Caches
mysql	variables	key_cache_block_size	long	The default size of key cache blocks
mysql	variables	key_cache_division_limit	long	The minimum percentage of warm blocks in key cache
mysql	variables	language	string	Client error messages in given language. May be given as a full path
mysql	variables	large_files_support	string	Whether large files are supported
mysql	variables	large_page_size	long	General.Features
mysql	variables	large_pages	string	Enable support for large pages
mysql	variables	last_insert_id	long	
mysql	variables	lc_time_names	string	General.Miscellaneous
mysql	variables	license	string	Type of license for the server
mysql	variables	local_infile	string	Enable/disable LOAD DATA LOCAL INFILE (takes values 1|0)
mysql	variables	locked_in_memory	long	Whether mysqld is locked in memory with --memlock
mysql	variables	log	string	Log connections and queries to file
mysql	variables	log_bin	string	Enables binary log
mysql	variables	log_bin_trust_function_creators	string	General.Logs
mysql	variables	log_bin_trust_routine_creators	string	(deprecated) Use log-bin-trust-function-creators
mysql	variables	log_error	string	Error log file
mysql	variables	log_output	string	
mysql	variables	log_queries_not_using_indexes	string	General.Logs
mysql	variables	log_slave_updates	string	General.Logs
mysql	variables	log_slow_queries	string	General.Logs
mysql	variables	log_update	string	General.Logs
mysql	variables	log_warnings	long	Log some not critical warnings to the log file

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	long_query_time	long	General.Performance
mysql	variables	low_priority_updates	string	INSERT/DELETE/UPDATE has lower priority than selects
mysql	variables	lower_case_file_system	string	General.Features
mysql	variables	lower_case_table_names	long	SQL.Overview
mysql	variables	max_allowed_packet	long	Max packetlength to send/receive from to server
mysql	variables	max_binlog_cache_size	long	General.Logs
mysql	variables	max_binlog_size	long	General.Logs
mysql	variables	max_connect_errors	long	Connections.Overview
mysql	variables	max_connections	long	The number of simultaneous clients allowed
mysql	variables	max_delayed_threads	long	General.Performance
mysql	variables	max_error_count	long	Max number of errors/warnings to store for a statement
mysql	variables	max_heap_table_size	long	Don't allow creation of heap tables bigger than this
mysql	variables	max_insert_delayed_threads	long	This variable is a synonym for max_delayed_threads
mysql	variables	max_join_size	long	General.Performance
mysql	variables	max_length_for_sort_data	long	Max number of bytes in sorted records
mysql	variables	max_prepared_stmt_count	long	Maximum number of prepared statements in the server
mysql	variables	max_relay_log_size	long	General.Logs
mysql	variables	max_seeks_for_key	long	Limit assumed max number of seeks when looking up rows based on a key
mysql	variables	max_sort_length	long	The number of bytes to use when sorting BLOB or TEXT values
mysql	variables	max_sp_recursion_depth	long	Maximum stored procedure recursion depth
mysql	variables	max_tmp_tables	long	Maximum number of temporary tables a client can keep open at a time
mysql	variables	max_user_connections	long	Connections.Overview
mysql	variables	max_write_lock_count	long	After this many write locks, allow some read locks to run in between
mysql	variables	min_examined_row_limit	long	
mysql	variables	multi_range_count	long	General.Performance
mysql	variables	myisam_data_pointer_size	long	Default pointer size to be used for MyISAM tables
mysql	variables	myisam_max_extra_sort_file_size	long	Deprecated option
mysql	variables	myisam_max_sort_file_size	long	Table Types.MyISAM
mysql	variables	myisam_recover_options	string	The value of the --myisam-recover option
mysql	variables	myisam_repair_threads	long	Table Types.MyISAM

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	myisam_sort_buffer_size	long	Memory.Buffers
mysql	variables	myisam_stats_method	string	Specifies how MyISAM index statistics collection code should treat NULLs
mysql	variables	myisam_use_mmap	string	
mysql	variables	named_pipe	string	Indicates whether the server supports connections over named pipes
mysql	variables	net_buffer_length	long	Buffer length for TCP/IP and socket communication
mysql	variables	net_read_timeout	long	Connections.Networking
mysql	variables	net_retry_count	long	Connections.Networking
mysql	variables	net_write_timeout	long	Connections.Networking
mysql	variables	new	string	Use very new, possibly 'unsafe' functions
mysql	variables	old	string	
mysql	variables	old_alter_table	string	
mysql	variables	old_passwords	string	Use old password encryption method (needed for 4.0 and older clients)
mysql	variables	open_files_limit	long	General.Directories / Files
mysql	variables	optimizer_prune_level	long	General.Features
mysql	variables	optimizer_search_depth	long	Maximum depth of search performed by the query optimizer
mysql	variables	pid_file	string	Pid file used by safe_mysqld
mysql	variables	plugin_dir	string	
mysql	variables	port	long	Port number to use for connection
mysql	variables	preload_buffer_size	long	The size of the buffer that is allocated when preloading indexes
mysql	variables	prepared_stmt_count	long	The current number of prepared statements
mysql	variables	protocol_version	long	The version of the client/server protocol used by the MySQL server
mysql	variables	pseudo_thread_id	long	
mysql	variables	query_alloc_block_size	long	Allocation block size for query parsing and execution
mysql	variables	query_cache_limit	long	Don't cache results that are bigger than this
mysql	variables	query_cache_min_res_unit	long	Memory.Caches
mysql	variables	query_cache_size	long	The memory allocated to store results from old queries
mysql	variables	query_cache_type	string	Query cache type
mysql	variables	query_cache_wlock_invalidate	string	Invalidate queries in query cache on LOCK for write
mysql	variables	query_prealloc_size	long	Persistent buffer for query parsing and execution
mysql	variables	rand_seed1	long	
mysql	variables	rand_seed2	long	
mysql	variables	range_alloc_block_size	long	Allocation block size for storing ranges during optimization
mysql	variables	read_buffer_size	long	Memory.Buffers

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	read_only	string	Make all nontemporary tables read only
mysql	variables	read_rnd_buffer_size	long	Memory.Buffers
mysql	variables	relay_log	string	
mysql	variables	relay_log_index	string	
mysql	variables	relay_log_info_file	string	
mysql	variables	relay_log_purge	string	Determines whether relay logs are purged
mysql	variables	relay_log_space_limit	long	Maximum space to use for all relay logs
mysql	variables	report_host	string	
mysql	variables	report_password	string	
mysql	variables	report_port	long	
mysql	variables	report_user	string	
mysql	variables	rpl_recovery_rank	long	Not used
mysql	variables	secure_auth	string	Disallow authentication for accounts that have old (pre-4.1) passwords
mysql	variables	secure_file_priv	string	Unknown.Unknown
mysql	variables	server_id	long	General.Overview
mysql	variables	shared_memory	string	Whether the server allows shared-memory connections
mysql	variables	shared_memory_base_name	string	The name of shared memory to use for shared-memory connections
mysql	variables	skip_external_locking	string	Skip system (external) locking
mysql	variables	skip_networking	string	Don't allow connection with TCP/IP
mysql	variables	skip_show_database	string	Don't allow 'SHOW DATABASE' commands
mysql	variables	slave_compressed_protocol	string	Use compression on master/slave protocol
mysql	variables	slave_exec_mode	string	
mysql	variables	slave_load_tmpdir	string	General.Replication
mysql	variables	slave_net_timeout	long	Connections.Networking
mysql	variables	slave_skip_errors	string	General.Replication
mysql	variables	slave_transaction_retries	long	General.Replication
mysql	variables	slow_launch_time	long	General.Performance
mysql	variables	slow_query_log	string	
mysql	variables	slow_query_log_file	string	
mysql	variables	socket	string	Socket file to use for connection
mysql	variables	sort_buffer_size	long	Each thread that needs to do a sort allocates a buffer of this size
mysql	variables	sql_auto_is_null	string	
mysql	variables	sql_big_selects	string	General.Features
mysql	variables	sql_big_tables	string	
mysql	variables	sql_buffer_result	string	
mysql	variables	sql_log_bin	string	
mysql	variables	sql_log_off	string	
mysql	variables	sql_log_update	string	

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	sql_low_priority_updates	string	
mysql	variables	sql_max_join_size	long	
mysql	variables	sql_mode	string	Set the SQL server mode
mysql	variables	sql_notes	string	If set to 1, warnings of Note level are recorded
mysql	variables	sql_quote_show_create	string	
mysql	variables	sql_safe_updates	string	
mysql	variables	sql_select_limit	long	
mysql	variables	sql_slave_skip_counter	long	
mysql	variables	sql_warnings	string	General.Miscellaneous
mysql	variables	ssl_ca	string	CA file in PEM format (check OpenSSL docs, implies --ssl)
mysql	variables	ssl_capath	string	CA directory (check OpenSSL docs, implies --ssl)
mysql	variables	ssl_cert	string	X509 certification in PEM format (implies --ssl)
mysql	variables	ssl_cipher	string	SSL cipher to use (implies --ssl)
mysql	variables	ssl_key	string	X509 key in PEM format (implies ssl)
mysql	variables	storage_engine	string	The default storage engine
mysql	variables	sync_binlog	long	Synchronously flush binary log to disk after every #th event
mysql	variables	sync_frm	string	Sync .frm to disk on create. Enabled by default
mysql	variables	sync_replication	long	Deprecated
mysql	variables	sync_replication_slave_id	long	Deprecated
mysql	variables	sync_replication_timeout	long	Deprecated
mysql	variables	system_time_zone	string	The server system time zone
mysql	variables	table_cache	long	Deprecated; use --table_open_cache instead
mysql	variables	table_definition_cache	long	
mysql	variables	table_lock_wait_timeout	long	General.Miscellaneous
mysql	variables	table_open_cache	long	
mysql	variables	table_type	string	A synonym for storage_engine
mysql	variables	thread_cache_size	long	How many threads we should keep in a cache for reuse
mysql	variables	thread_handling	string	
mysql	variables	thread_stack	long	The stack size for each thread
mysql	variables	time_format	string	The TIME format (for future)
mysql	variables	time_zone	string	The current time zone.
mysql	variables	timed_mutexes	string	General.Miscellaneous
mysql	variables	timestamp	long	
mysql	variables	tmp_table_size	long	Memory.Overview
mysql	variables	tmpdir	string	Path for temporary files

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	transaction_alloc_block_size	long	Allocation block size for transactions to be stored in binary log
mysql	variables	transaction_prealloc_size	long	Persistent buffer for transactions to be stored in binary log
mysql	variables	tx_isolation	string	The default transaction isolation level
mysql	variables	unique_checks	string	
mysql	variables	updatable_views_with_limit	string	SQL.Overview
mysql	variables	version	string	Output version information and exit
mysql	variables	version_comment	string	General.Overview
mysql	variables	version_compile_machine	string	The type of machine or architecture on which MySQL was built
mysql	variables	version_compile_os	string	The type of operating system on which MySQL was built
mysql	variables	wait_timeout	long	Connections.Overview
mysql	variables	warning_count	long	
mysql	waiting_db	waiting_db	string	
mysql	waiting_statement	waiting_statement	string	
mysql	waiting_thread	waiting_thread	long	
mysql	waiting_time	waiting_time	long	
mysql	waiting_user	waiting_user	string	
os	cpu	cpu_cache_size	long	The size of the CPU cache
os	cpu	cpu_idle	long_counter	The CPU idle time
os	cpu	cpu_mhz	long	The MHz rating of the CPU
os	cpu	cpu_name	string	The CPU name
os	cpu	cpu_sys	long_counter	CPU system / kernel time
os	cpu	cpu_user	long_counter	CPU user space time
os	cpu	cpu_vendor	string	The name of the CPU vendor
os	cpu	cpu_wait	long_counter	CPU IO wait time
os	disk	disk_bytes_read	long_counter	The number of bytes read from disk
os	disk	disk_bytes_written	long_counter	The number of bytes written to disk
os	disk	disk_queue	long	The size of the disk IO queue
os	disk	disk_reads	long_counter	The number of disk reads
os	disk	disk_rtime	long	
os	disk	disk_servicetime	double	
os	disk	disk_snaptime	long	
os	disk	disk_time	long	
os	disk	disk_writes	long_counter	The number of disk writes
os	disk	disk_wtime	long	
os	disk	fs_avail	long	The amount of available space for the mount point
os	disk	fs_files	long	The number of files on the mount point
os	disk	fs_free_files	long	The number of free files on the mount point
os	disk	fs_total	long	The total amount of space for the mount point
os	disk	fs_unused	long	The amount of unused space for the

Namespace	Namespace Type	Attribute	Type	Description
				mount point
os	disk	fs_used	long	The amount of used space for the mount point
os	fs	fs_avail	long	
os	fs	fs_dev_name	string	
os	fs	fs_dir_name	string	
os	fs	fs_files	long	
os	fs	fs_flags	long	
os	fs	fs_free_files	long	
os	fs	fs_options	string	
os	fs	fs_sys_type_name	string	
os	fs	fs_total	long	
os	fs	fs_type_name	string	
os	fs	fs_unused	long	
os	fs	fs_used	long	
os	Host	host_id	string	The host unique identifier
os	Host	name	string	
os	loadavg	0	double	
os	loadavg	1	double	
os	loadavg	2	double	
os	mem	ram_total	long	The total amount of available RAM
os	mem	ram_unused	long	The total amount of unused RAM
os	mem	swap_page_in	long	
os	mem	swap_page_out	long	
os	mem	swap_total	long	The total amount of available swap memory
os	mem	swap_unused	long	The total amount of unused swap memory
os	net	address	string	The address of the network interface
os	net	broadcast	string	The network interface broadcast address
os	net	description	string	The network interface description
os	net	destination	string	The network interface destination address
os	net	flags	long	The network interface flags
os	net	hwaddr	string	The network interface MAC address
os	net	metric	long	Operating System.Net
os	net	mtu	long	The Maximum Transmission Unit
os	net	name	string	The network interface name (eth0, eth1 etc.)
os	net	netmask	string	The network interface subnet mask
os	net	rx_bytes	long_counter	The number of bytes received
os	net	rx_dropped	long_counter	The number of received packets dropped
os	net	rx_errors	long_counter	The number of received packet errors
os	net	rx_frame	long_counter	The number received packet frame errors

Namespace	Namespace Type	Attribute	Type	Description
os	net	rx_overruns	long_counter	The number of received packet overruns
os	net	rx_packets	long_counter	The number of packets received
os	net	speed	long	The network interface connection speed
os	net	tx_bytes	long_counter	The number of bytes transmitted
os	net	tx_carrier	long_counter	The number of carrier errors on network transmission
os	net	tx_collisions	long_counter	The number of network transmission collisions
os	net	tx_dropped	long_counter	The number of network transmissions dropped
os	net	tx_errors	long_counter	The number of network transmission errors
os	net	tx_overruns	long_counter	The number of network transmission overruns
os	net	tx_packets	long_counter	The number of packets transmitted
os	net	type	string	The network interface type (ethernet, loopback etc.)
os	os	os_arch	string	The architecture of the OS / CPU
os	os	os_description	string	Description of the OS
os	os	os_machine	string	Machine CPU architecture
os	os	os_name	string	The OS name
os	os	os_patchlevel	string	The OS patch level
os	os	os_vendor	string	The OS vendor
os	os	os_vendor_code_name	string	The OS vendor code name
os	os	os_vendor_name	string	The OS vendor name
os	os	os_vendor_version	string	The OS vendor version number
os	os	os_version	string	The OS version number
util	LogHistogram	0	long_counter	
util	LogHistogram	1	long_counter	
util	LogHistogram	10	long_counter	
util	LogHistogram	11	long_counter	
util	LogHistogram	12	long_counter	
util	LogHistogram	13	long_counter	
util	LogHistogram	14	long_counter	
util	LogHistogram	15	long_counter	
util	LogHistogram	16	long_counter	
util	LogHistogram	17	long_counter	
util	LogHistogram	18	long_counter	
util	LogHistogram	19	long_counter	
util	LogHistogram	2	long_counter	
util	LogHistogram	20	long_counter	
util	LogHistogram	21	long_counter	
util	LogHistogram	22	long_counter	
util	LogHistogram	23	long_counter	
util	LogHistogram	24	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
util	LogHistogram	25	long_counter	
util	LogHistogram	26	long_counter	
util	LogHistogram	27	long_counter	
util	LogHistogram	28	long_counter	
util	LogHistogram	29	long_counter	
util	LogHistogram	3	long_counter	
util	LogHistogram	30	long_counter	
util	LogHistogram	31	long_counter	
util	LogHistogram	32	long_counter	
util	LogHistogram	33	long_counter	
util	LogHistogram	34	long_counter	
util	LogHistogram	35	long_counter	
util	LogHistogram	36	long_counter	
util	LogHistogram	37	long_counter	
util	LogHistogram	38	long_counter	
util	LogHistogram	39	long_counter	
util	LogHistogram	4	long_counter	
util	LogHistogram	40	long_counter	
util	LogHistogram	41	long_counter	
util	LogHistogram	42	long_counter	
util	LogHistogram	43	long_counter	
util	LogHistogram	44	long_counter	
util	LogHistogram	45	long_counter	
util	LogHistogram	46	long_counter	
util	LogHistogram	47	long_counter	
util	LogHistogram	48	long_counter	
util	LogHistogram	49	long_counter	
util	LogHistogram	5	long_counter	
util	LogHistogram	50	long_counter	
util	LogHistogram	51	long_counter	
util	LogHistogram	52	long_counter	
util	LogHistogram	53	long_counter	
util	LogHistogram	54	long_counter	
util	LogHistogram	55	long_counter	
util	LogHistogram	56	long_counter	
util	LogHistogram	57	long_counter	
util	LogHistogram	58	long_counter	
util	LogHistogram	59	long_counter	
util	LogHistogram	6	long_counter	
util	LogHistogram	60	long_counter	
util	LogHistogram	61	long_counter	
util	LogHistogram	62	long_counter	
util	LogHistogram	7	long_counter	
util	LogHistogram	8	long_counter	
util	LogHistogram	9	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
util	LogHistogram	`base`	`long`	
util	LogHistogram	`nul`	`long_counter`	