MySQL Enterprise Monitor 2.2

MySQL Enterprise Monitor 2.2 Manual

Abstract

This manual documents the MySQL Enterprise Monitor version 2.2.

Document generated on: 2010-11-03 (revision: 2005)

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 2.2. 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. MySQL Enterprise Monitor Introduction and Architecture

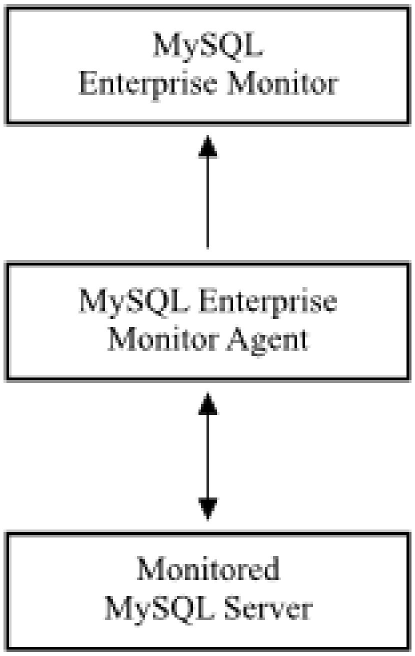
MySQL Enterprise Monitor is an enterprise monitoring system for MySQL that keeps an eye on your MySQL servers, notifies you of potential issues and problems, and advises you how to fix the issues. MySQL Enterprise Monitor can monitor all kinds of configurations, from a single MySQL server that is important to your business, all the way up to a huge farm of MySQL servers powering a busy web site.

This section describes the basic components that make up the MySQL Enterprise Monitor product. These components can be installed in various configurations depending on your own database and network topology, to give you the best combination of reliable and responsive monitoring data, with minimal overhead on the database server machines. A typical MySQL Enterprise Monitor installation consists of:

- One or more MySQL servers that you want to monitor. MySQL Enterprise Monitor can monitor both Community and Enterprise MySQL server releases.
- A MySQL Enterprise Agent for each monitored MySQL server.
- A single MySQL Enterprise Service Manager, which collates information from the agents, and provides the user interface to the collected data.

The structure of a typical MySQL Enterprise Monitor installation is shown in the figure below.

Figure 1.1. MySQL Enterprise Monitor Architecture



MySQL Enterprise Monitor is designed to monitor one or more MySQL servers. The monitoring information is collected by using an agent, *MySQL Enterprise Agent*. The agent communicates with the MySQL server that it monitors, collecting variables, status and health information, and sending this information to the MySQL Enterprise Service Manager. If you have multiple MySQL servers, then you have multiple MySQL Enterprise Agent processes monitoring each MySQL server. The MySQL Enterprise Agent has other responsibilities and capabilities, which we describe in more detail later in this chapter.

The information collected by the agent about each MySQL server you are monitoring is sent to the *MySQL Enterprise Service Manager*. This server collates all of the information from the agents. As it collates the information sent by the agents, the MySQL Enterprise Service Manager continually tests the collected data, comparing the status of the server to reasonable values. When thresholds are reached, the server can trigger an event (including an alarm and notification) to highlight a potential issue, such as low memory, high CPU usage, or more complex conditions such insufficient buffer sizes and status information. We call each test, with its associated threshold value, a *rule*.

These rules, and the alarms and notifications, are each known as a *MySQL Enterprise Advisor*. Advisors form a critical part of the MySQL Enterprise Service Manager, as they provide warning information and troubleshooting advice about potential problems.

The MySQL Enterprise Service Manager includes a web server, and you interact with it through any web browser. This interface, the MySQL Enterprise Dashboard, displays all of the information collected by the agents, and lets you view all of your servers and their current status as a group or individually. You control and configure all aspects of the service using the MySQL Enterprise Dashboard.

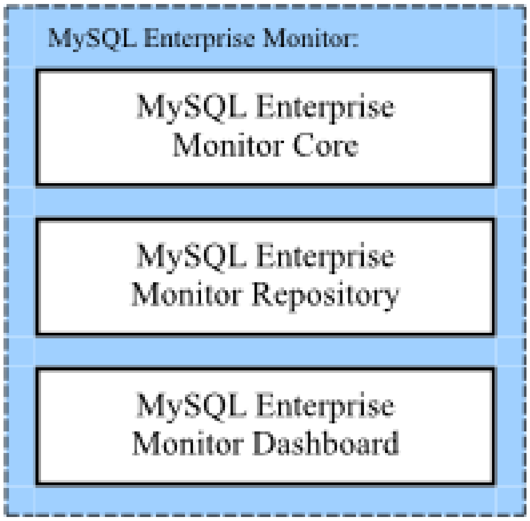
The information supplied by the MySQL Enterprise Agent processes also includes statistical and query information, which you can view in the form of graphs. For example, you can view aspects such as server load, query numbers, or index usage information as a graph over time. The graph lets you pinpoint problems or potential issues on your server, and can help diagnose the impact from database or external problems (such as external system or network failure) by examining the data from a specific time interval.

You can configure the MySQL Enterprise Agent to collect detailed information about the queries executed on your server, including the row counts and performance times for executing each query. You can correlate the detailed query data with the graphical information to identify which queries were executing when you experienced a particularly high load, index or other issue. The query data is supported by a system called Query Analyzer, and the data can be presented in different ways depending on your needs.

1.1. MySQL Enterprise Service Manager

The MySQL Enterprise Service Manager is the hub of MySQL Enterprise Monitor: it collects information, stores the information for display, generating warnings and alerts, sends emails and other notifications, and supports the MySQL Enterprise Dashboard as the interface to the system.

Figure 1.2. The MySQL Enterprise Service Manager Architecture


MySQL Enterprise Monitor:
MySQL Enterprise Monitor Core
MySQL Enterprise Monitor Repository
MySQL Enterprise Monitor Dashboard

You can consider the MySQL Enterprise Service Manager as being composed of three parts:

- The MySQL Enterprise Service Manager core, built on top of the Tomcat web server, is the server that handles tasks such as collecting information from each MySQL Enterprise Agent, formatting and inserting information into the repository, and executing the rules from the MySQL Enterprise Advisor collections.

  Because the system is built on top of Tomcat and Java, the performance of the system depends on the configuration and optimization of the Java environment. The standard configuration is designed to support most typical installations, but for installations that involve a larger number of monitored servers, or that support a large numbers of clients using the MySQL Enterprise Dashboard, you can change the configuration to optimize performance. For example, you might increase the Java memory sizes for better caching.

- The data collected by the MySQL Enterprise Service Manager is stored within the MySQL Enterprise Monitor repository. The repository uses a MySQL database, which enables MySQL Enterprise Monitor to take advantage of MySQL features such as the InnoDB storage engine and partitioning systems.

  You can optimize the configuration of the MySQL server depending on the hardware capacity, for example extending the cache memory for the InnoDB storage engine or the thread processes. The default configuration places both the MySQL server and the Tomcat service on the same machine. To dedicate more resources to the MySQL and Tomcat components, you can separate the two across two different machines.

- The MySQL Enterprise Dashboard is a series of JSP pages displayed through the Tomcat service. These web pages display and control all the components of the system, including the different views of data, notification system, graphing engine and the configuration of the MySQL Enterprise Service Manager.

1.2. MySQL Enterprise Dashboard

The MySQL Enterprise Dashboard is a web-based interface to the MySQL Enterprise Service Manager. MySQL Enterprise Dashboard provides a quick overview of the current status of your server, and lets you drill down into the current status, events, and historical information submitted by each MySQL Enterprise Agent.

The main features of the MySQL Enterprise Dashboard include:

- A single page, **Monitor**, that gives an overview of the current health and status of all your servers, an overview of any events that are triggered, a heat chart for critical status information, and a customizable set of graphs.
- The **Heat Chart** table shows the current status of all your servers, using icons to highlight important issues. A heat chart is included on the main **Monitor** tab, and you can also keep a standalone **Heat Chart** open on your desktop to monitor your server states.
- The **Advisors** tab lets you create, edit, control, install, and schedule advisors within the system.
- The **Query Analyzer** helps you identify problem queries.
- The **Replication** view monitors the structure and health of your replication environment.
- The **Settings** tab lets you set the parameters for the server, including setting notification lists, configuring SNMP, and setting the intervals for removing old data.
- The **What's New?** tab gives a live connection to the MySQL Enterprise website, with news about the latest releases, critical fixes and patches, and any current support issues.

For a more detailed overview of MySQL Enterprise Dashboard, see Section 2.1, “MySQL Enterprise Dashboard”.

1.3. MySQL Enterprise Advisor Collections

The MySQL Enterprise Advisors are organized into collections, according to the subscription level of your MySQL Enterprise subscription. These collections include advisors and definitions for graphs that you can view within the MySQL Enterprise Dashboard.

Each MySQL Enterprise Advisor uses the information collected by the MySQL Enterprise Agent on each machine to test data values or configuration settings and send an event notification when a threshold is reached. Advisors can be a simple piece of advice, a warning, or a critical event. Within the MySQL Enterprise Service Manager these events are displayed through either the events tab, or for critical items, a heat chart summarizing the current status of your MySQL servers.

The events can be displayed within the MySQL Enterprise Dashboard, or can trigger additional notifications, such as an email or an SMS text message. The information provided by the advisors, and the collected data, are also exported through SNMP (Simple Network Management Protocol), so you can use SNMP traps within your existing network monitoring infrastructure.

All advisors are executed according to a *schedule* that determines how frequently each advisor is executed.

1.4. Monitored MySQL Server

MySQL Enterprise Monitor can monitor any MySQL server from MySQL 4.1 to 6.0. The advisors and rules provide specific support for different versions. For example, there are unique rules and graphs for different MySQL versions, and some rules provide advice about MySQL upgrades.

In addition to monitoring the internal operation of the MySQL server, the MySQL Enterprise Agent and MySQL Enterprise Service Manager can determine the replication topology of your MySQL servers and monitor the replication status.

MySQL Enterprise Monitor uses a number of unique values to identify the different components, including the MySQL instance being monitored:

- Each MySQL Server has a UUID, stored in the `mysql.inventory` table, that uniquely identifies the MySQL server to the rest of MEM. The server UUID is used to collate information about a single MySQL instance.
- Each host (the machine on which the agent is running) has a UUID to uniquely identify the host to the rest of MySQL Enterprise Monitor. This is used to collate the OS information (such as CPU, RAM and disk data). The hostid also determines whether the MySQL server is on the same host as it was before, to identify when data has been moved between machines, or when a machine has been upgraded. The host UUID is stored within the `hostid` row within the `mysql.inventory` table.
- Each agent has a UUID to identify the agent to MEM. The agent UUID is defined within the `agent-uuid` parameter within the agent configuration file.

These UUIDs are used in combination to register and collate information, and to determine the location and source of an issue.

You can monitor any MySQL server, because the agent communicates with the MySQL server using the same protocol as any other client. For best results when using MySQL Enterprise Agent with your MySQL Server, follow these guidelines:

- Each instance must have a unique ID, as stored within the `mysql.inventory` table.
- Configure the MySQL server with a user that has the correct permissions and rights to be able to monitor the different parameters of the server. The installation guide provides instructions to create a suitable user.

1.5. MySQL Enterprise Agent

The MySQL Enterprise Agent performs the following tasks:

- Connects to, and collects, the core information about a MySQL server. The core data is called the *inventory*. The inventory includes critical information (known items) about the server which is used to help determine the other information that can be collected. For example, the inventory includes details such as the MySQL server version number, supported storage engines and replication configuration.
- Collects a set of standard data items that constitute the bulk of the data sent to the MySQL Enterprise Service Manager. This includes all of the information about the configuration, database objects and other content.
- Collects various pieces of information about the host operating system on which the agent is running. This includes CPU, RAM and filesystem information and statistics.
- Collects query analyzer data by acting as a MySQL Proxy, collecting, and forwarding MySQL queries from clients and sending the collected query execution information to the MySQL Enterprise Service Manager. This is known as the MySQL Enterprise Agent Proxy Service.
- Collects any custom data items that you can optionally configure to provide unique monitoring and statistical information for your MySQL server.

> **Note**
>
> Although the MySQL Enterprise Agent consists of a number of different elements, the components are used collectively as a single application, the MySQL Enterprise Agent. However, in order to distinguish these different elements and their

function, they are individually reference and recognized throughout the documentation so that their functionality, configuration, and deployment can be more easily described.

The agent is a general purpose service that provides the basis for all this functionality and is the key component of MySQL Enterprise Monitor as it provides the MySQL Enterprise Service Manager with the information about each individual MySQL server.

Each MySQL Enterprise Agent requires a UUID to uniquely identify the MySQL Enterprise Agent to the MySQL Enterprise Service Manager. This UUID is independent of the UUID assigned to each monitored server. Take care that you do not copy the UUID from one instance of the agent to another, as it will cause incorrect recording of the information by the agent.

When the agent first starts, it performs the following operations:

- Loads the configuration information.
- Checks that the Agent UUID exists.
- Generates what it thinks the current Host ID should be.
- Starts the network thread and collector threads.
- Compares whether the instance we are connected to has the same Host ID, to make sure we are not starting on a copied instance wrongly.
- Registers with the MySQL Enterprise Service Manager and then sends a heartbeat signal.
- Receives a "list known data items" call from the Service Manager.
- Loads the list of data items, including any custom data items, to be read from the server.
- Responds to the Service Manager with the known items.
- Receives a "list instances" call from the Service Manager for certain data items.
- Runs the inventory process to determine the core information about the MySQL server.
- Responds to the Service Manager with the inventory.
- Receives schedule information for each data item from the MySQL Enterprise Service Manager in "collect data" tasks.
- Starts collecting data and sending it to the MySQL Enterprise Service Manager according to the schedule set during configuration.

Once running, each MySQL Enterprise Agent periodically collects information from the monitored MySQL server and sends it to MySQL Enterprise Agent.

1.5.1. MySQL Enterprise Agent Basic Architecture

The MySQL Enterprise Agent core architecture provides the basic chassis on which the other components operate. The main purpose of the core architecture is to support the connectivity to the MySQL server being monitored, and the communication ability with the MySQL Enterprise Service Manager to collect and send collected information, and receive configuration and other information from MySQL Enterprise Service Manager

In addition to providing the main chassis for the MySQL Enterprise Agent Proxy Service component, the agent also performs a number of key operations in collecting information from the MySQL server that is being monitored. In particular, the core agent:

- Provides the direct monitoring of the MySQL server, including checking the server accessibility, configuration, obtaining the server ID, and setting up the environment to start collecting more detailed information. In addition to the information accessible by accessing variable and configuration information within the server, other configuration parameters, such as the replication configuration, is also recovered from the server.
- Collects the operating system specific information, including RAM, disk storage and other data.
- Collects the data from the server, including obtaining the values and configuration of the MySQL server, status variables and other information.

- Communicates with the MySQL Enterprise Service Manager. Data is collected at scheduled intervals according to the schedule set by the server. This information is then sent to the MySQL Enterprise Service Manager at a regular schedule that is designed reduce the effect of multiple agents on multiple machines simultaneously sending large blocks of data up to the MySQL Enterprise Service Manager at the same time.

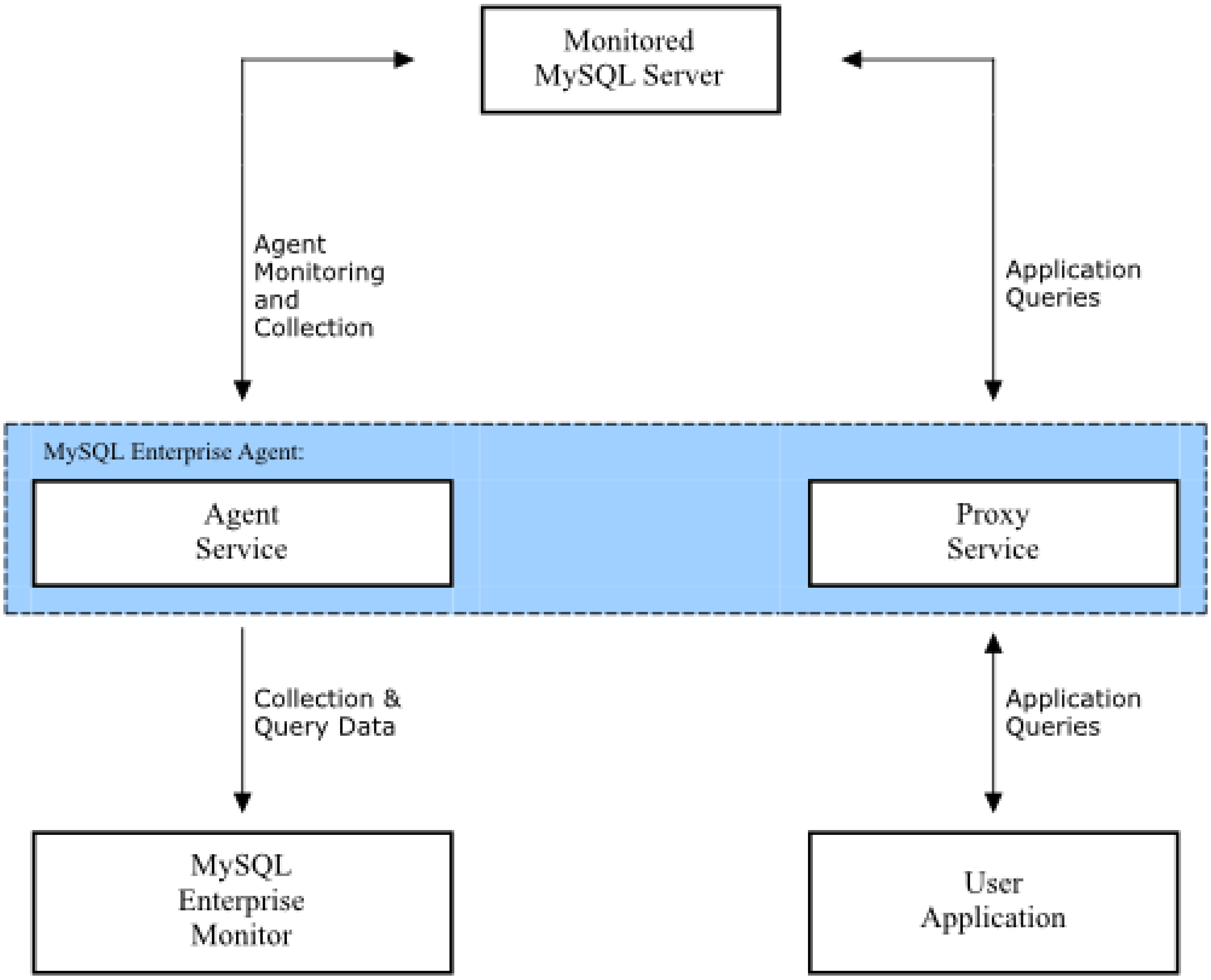
These responsibilities are handled in addition to the MySQL Enterprise Agent Proxy Service service that provide information about queries for the query analyzer.

Because the MySQL Enterprise Agent and the MySQL Enterprise Service Manager may lose communication with each other (for example, due to transient network errors), the MySQL Enterprise Agent can store the data that would normally be sent up to the MySQL Enterprise Service Manager. Once communication is re-established, the saved data is sent up to the server.

1.5.2. MySQL Enterprise Agent Proxy Service Architecture

When using MySQL Enterprise Agent as a proxy service, the queries from user applications are sent through the MySQL Enterprise Agent, which forwards them onto a MySQL server, returning the query results while recording query statistical information. When operating in this mode, the agent acts as both the agent and proxy for the same monitored MySQL server (see Figure 1.3, "The MySQL Enterprise Monitor Architecture when using proxy and agent services for the same MySQL Server").

Figure 1.3. The MySQL Enterprise Monitor Architecture when using proxy and agent services for the same MySQL Server



The figures shows a single MySQL Enterprise Agent and two of the components, the core agent service and the MySQL Enterprise Agent Proxy Service. The MySQL Enterprise Agent is connected to the single monitored MySQL server. In turn, the MySQL Enterprise Agent is connected to the MySQL Enterprise Service Manager to provide monitoring and query analyzer data, and to the user application, which uses the proxy service to supply query analyzer data.

Chapter 2. Using and Configuring MySQL Enterprise Dashboard

This chapter contains information about the main sections of MySQL Enterprise Dashboard, how to use the interface, and how to set the different configuration and other settings.

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 `https://host:18443`. You can override these values during installation. To check the configured values, look at the `configuration_report.txt` file within your installation directory.

You can configure additional users to have access to your MySQL Enterprise Dashboard. Each user can have a specific privilege level: read-only, restricted from using the Query Analyzer data, or prevented from using any administrator features. For more information, see Section 2.8.4, "Managing Users".

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

2.1. MySQL Enterprise Dashboard

The MySQL Enterprise Dashboard provides you with information about your MySQL servers: 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.

You access the MySQL Enterprise Dashboard through a standard web browser. For more details on the supported browsers, see Section B.2, "Supported Browsers".

The MySQL Enterprise Dashboard page is divided into four sections: the header, the footer, the main bar, and an context-sensitive **Servers** list. These areas perform the following functions:

- **The Header**

  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.

  The **Refresh** selection configures the MySQL Enterprise Dashboard page to automatically refresh and reload at a specific interval, for pages that you leave open for a long time and consult at a glance. Do not make the interval too short, or the load on the MySQL Enterprise Service Manager might affect overall performance.

- **The Footer**

  The footer has external links to MySQL Enterprise and information about the current user. You can remain connected to the Dashboard and update your 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 2.8.7, "Product Information".

- **The Server Tree**

  The tab on the left displays the Server tree, where you navigate to different groups or to specific servers. 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.

  When you change your server selection, the information shown in the graphs and in the **Heat Chart** changes.

  > **Note**

For more information about server groups, see Section 2.8.3.2, “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 Main Section**

  The main section of the page shows the relevant information for the selected task or option. For example, when the **QUERY ANALYZER** tab is selected, the main section of the page shows the Query Analyzer data and a list of the current queries.

You can navigate to a number of pages that provide more detailed information. These pages include:

- **MONITOR**: The overview page that shows 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 lets you schedule their execution on different servers, apply and manage rules and manage the advisor installation itself. For more information, see Section 2.3, “The **ADVISORS** Tab”.
- **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 Section 2.4, “The **EVENTS** Tab”.
- **GRAPHS**: Lets you view and configure graphs covering various statistics. For details on how to view and use these graphs, see Section 2.5, “The **GRAPHS** Tab”.
- **QUERY ANALYZER**: Interfaces to the query monitoring system that monitors and tracks the individual queries that are being executed on a system. You can find problem queries that may need optimization or that may affect server load. For more information, see Section 2.6, “The **QUERY ANALYZER** Tab”.
- **REPLICATION**: Provides information on the status and structure of your servers that use replication. This page is only available if you have a suitable subscription level. For more information, see Section 2.7, “The **REPLICATION** Tab”.
- **SETTINGS**: Controls the settings for the server, including email configuration, passwords, and server and user management. For more information, see Section 2.8, “The **SETTINGS** Tab”.
- **WHAT'S NEW?**: Provides the latest news and information about your installation directly from the Enterprise website. For more information on the data provided, Section 2.9, “The **WHAT'S NEW** Tab”.

2.2. The MONITOR Tab

The default tab, **MONITORING**, which is shown when initially logging in to MySQL Enterprise Dashboard, provides a quick overview of your current monitoring information, and provides an instant health check for all of the MySQL servers across the enterprise.

From this page, you can:

- View monitoring data and all critical MySQL Advisor Rule violations for all or selected servers.
- Close and annotate MySQL Advisor Rule violations.
- Determine if there is a Monitor Agent that is not communicating with the Service Manager.
- 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 defaults to a thumbnail view, but opens into a larger image upon being clicked.

Graphs are shown in the center of the page beneath the tabs. If applicable, you also see 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 can open the **HEAT CHART** in its own window by clicking the `Standalone Heat Chart` link. If applicable, you also see a list of critical events.

The `Show/Hide Legend` link toggles display of the explanations of 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, back up the originals and then copy and paste the alternate set into the `images` directory.

When a specific server is selected in the `Server Tree`, the `Meta Info` area beneath the legend shows details about this server. 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.

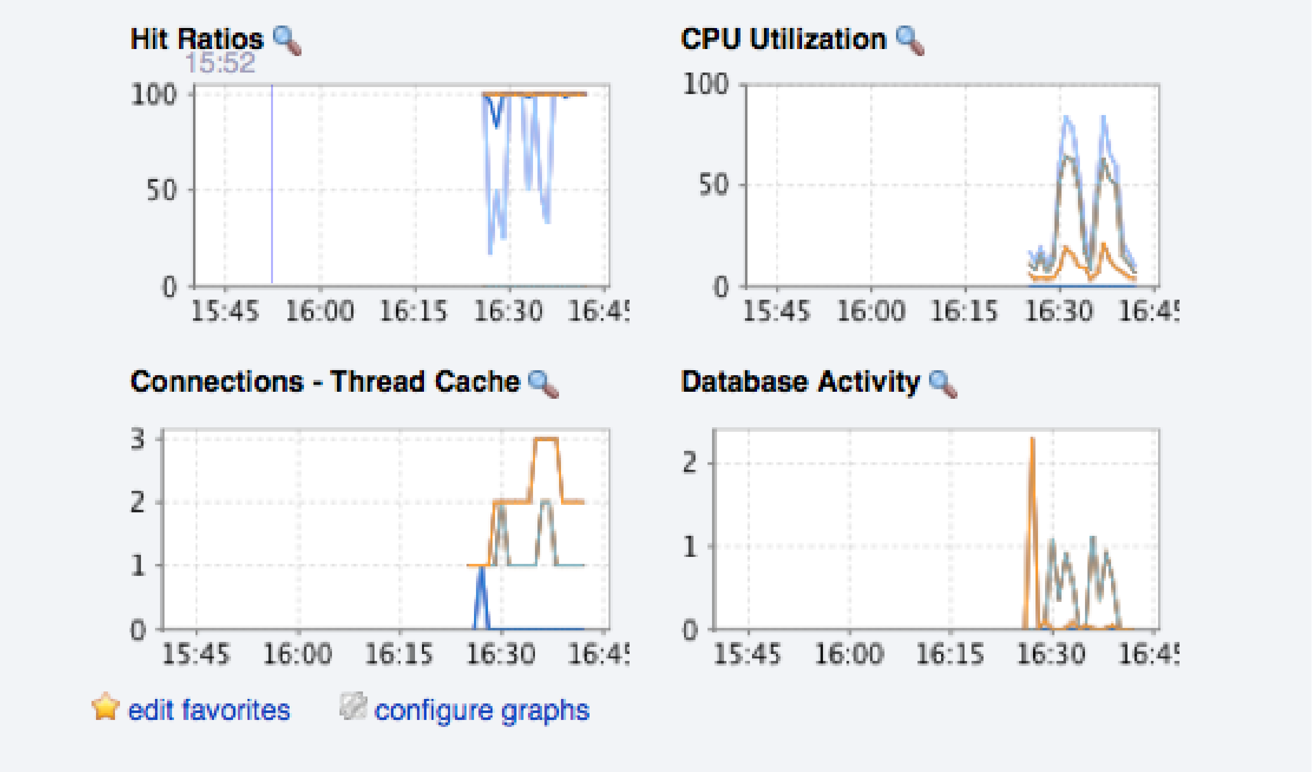
The `Meta Info` section also shows how long the agent has been running, when it last contacted the MySQL server it is monitoring, and the last time the agent contacted the dashboard. When you mouse over the date shown beside **UP SINCE**, a pop-up box displays the elapsed time since the server instance 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 5.3.7, “Configuring an Agent to Monitor a Remote MySQL Server”.

2.2.1. The Server Graphs and Critical Events

The center of the **MONITOR** page gives a visual representation of the state of your servers.

Figure 2.1. 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.

Graphs also let you drill down from overall usage information to the queries that were being executed on the server at the time. Click and drag on a graph to zoom in on the graph content and view the queries that were being executed during the specified period, to understand server performance and responsiveness. This information is only available when you configure query analysis, and enable query analysis for the servers you are monitoring. For more information, see Section 2.6, “The **Query Analyzer** Tab”.

To set the interval, click the `configure graphs` link immediately below the graphs. In the resulting dialog box, 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. You can also adjust the thumbnail and full-size graph dimensions from this dialog box. Save any changes, and the chosen values become the defaults whenever you log in.

You can also choose the default graphs shown on the **Monitor** page. 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. To select graphs that are not in a sequence, click the desired graphs while holding down the **Ctrl** key. The **Monitor** page can display a maximum of six graphs. Save your changes, and these become 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 `INSERT` statements.

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.

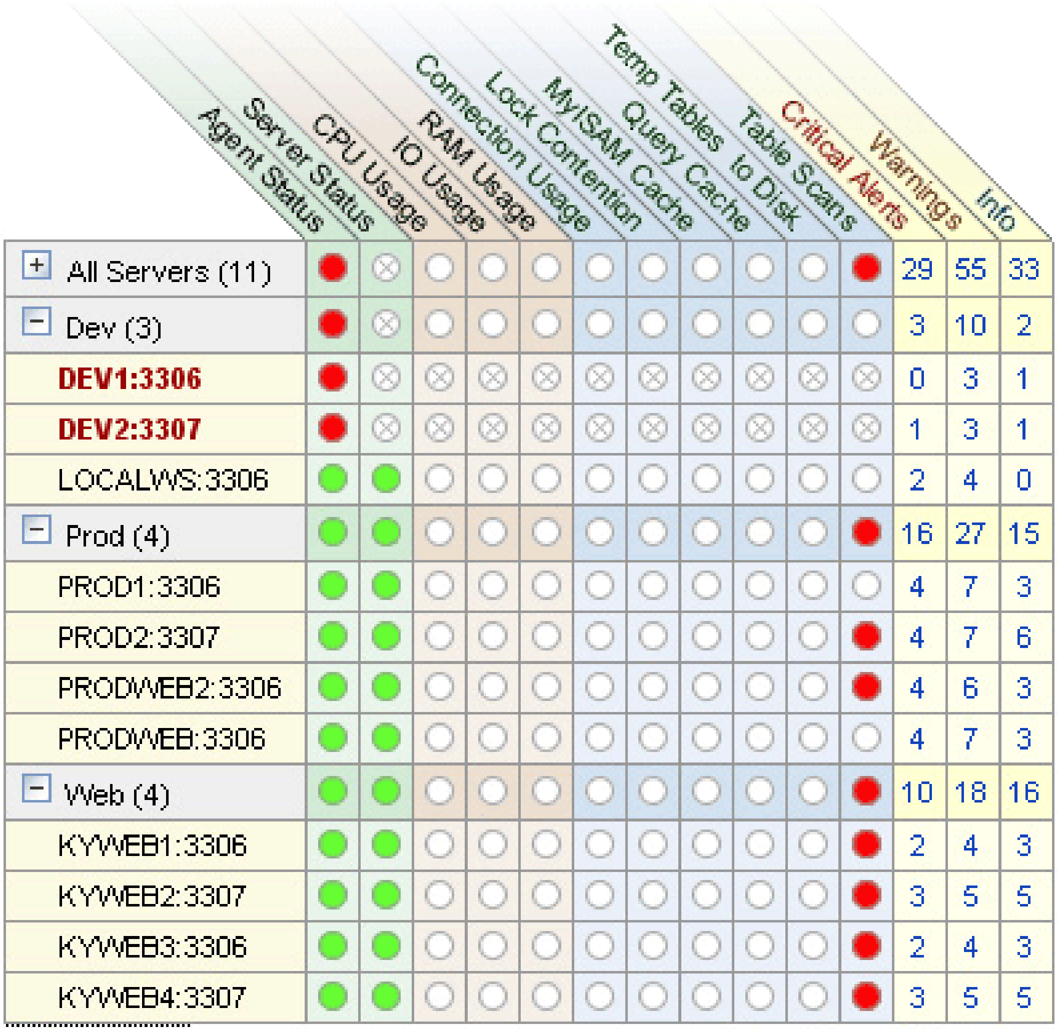
Alerts from the event system are displayed immediately below the charts. For more information on the events display, see Section 2.4, “The **Events** Tab”.

2.2.2. The Heat Chart

The **Heat Chart** is 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.

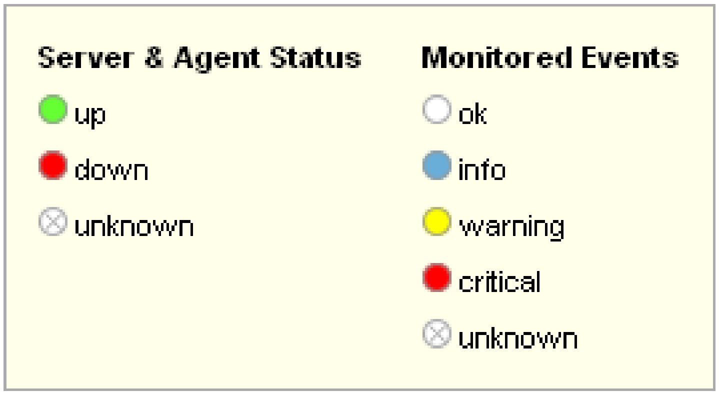
Figure 2.2. MySQL Enterprise Dashboard: The Heat Chart



	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 2.3. MySQL Enterprise Dashboard: The Heat Chart Legend



Server & Agent Status	Monitored Events
🟢 up	○ ok
🔴 down	🔵 info
⊗ unknown	🟡 warning
	🔴 critical
	⊗ unknown

The status `unknown` typically applies when an agent is down and can no longer report the status of the server that it is monitoring. The status `unknown` can also apply if the data that should be collected is not available on the server being monitored.

You can open the Heat Chart in its own browser window by clicking the **STANDALONE HEAT CHART** link immediately below the **HEAT CHART** on the left. You can set the refresh rate 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 **EVENTS** tab, which gives more detailed information. For more information about events, see Section 2.4, “The **EVENTS** Tab”. Hovering over the heat chart also provides server counts within the tooltips pop-up.

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 5.2.7.3, “Installing, Updating or Upgrading Advisors After Initial Login” and Section 2.8.5, “Manage Notification Groups”.

2.3. The ADVISORS Tab

> **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 via the Service Manager and the Monitor Agents, analyze that information based on custom rules included as part of MySQL Enterprise Monitor, 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.

The file `items-mysql-monitor.xml` contains supplementary information used to construct rules. On Windows, this file is 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.

2.3.1. Installing and Updating Advisors

Instructions for installing Advisors are given in Section 5.2.7.3, “Installing, Updating or Upgrading Advisors After Initial Login”, 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** tab 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 5.2.7.3, “Installing, Updating or Upgrading Advisors After Initial Login”.

2.3.2. 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.

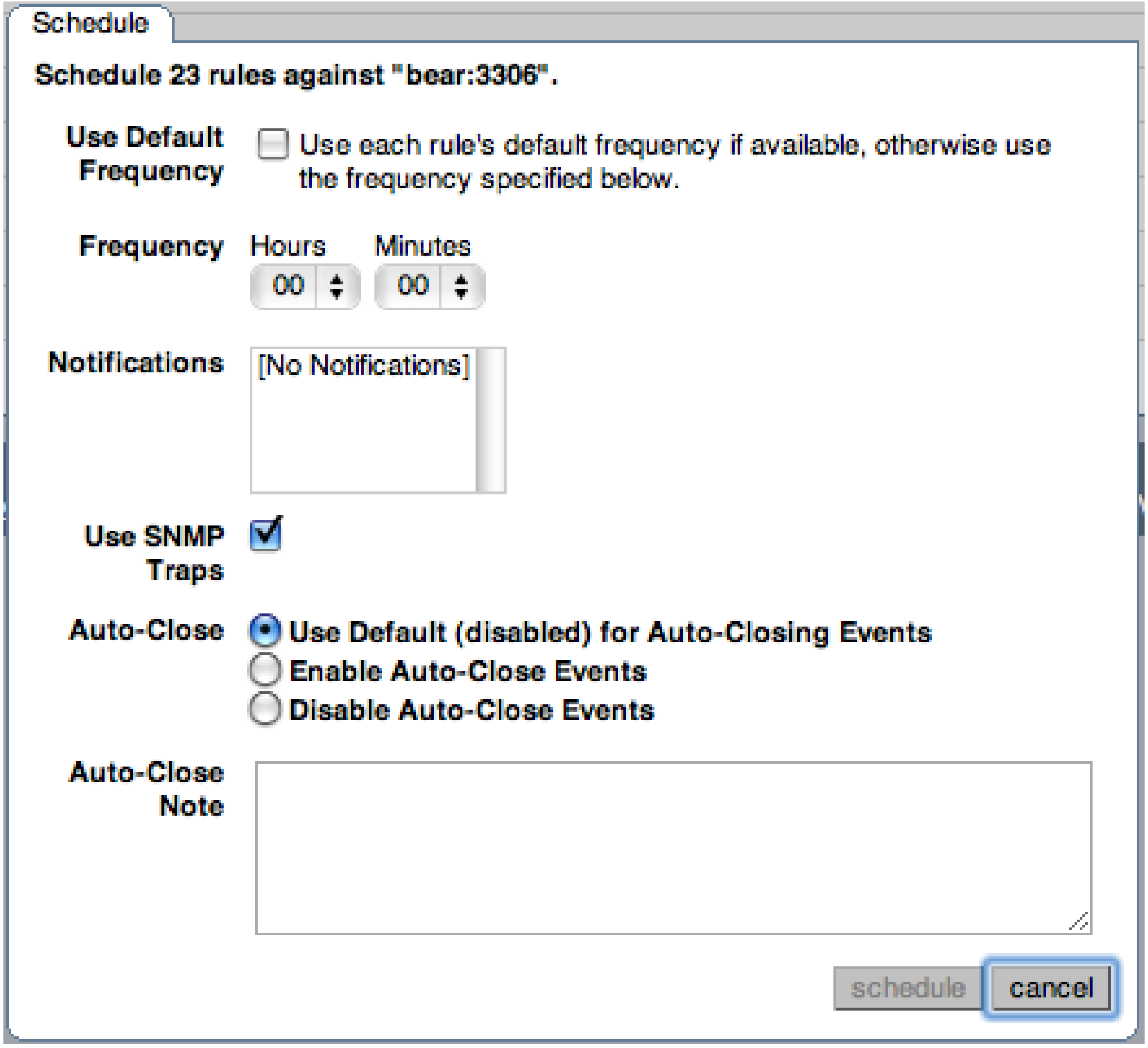
To view the advisors other than the Heat Chart group, select the `Add to Schedule` link. This will show all the advisors available for your subscription level.

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 2.4. MySQL Enterprise Dashboard: Scheduling Dialog



Schedule

Schedule 23 rules against "bear:3306".

Use Default Frequency ☐ Use each rule's default frequency if available, otherwise use the frequency specified below.

Frequency Hours 00 Minutes 00

Notifications [No Notifications]

Use SNMP Traps ☑

Auto-Close
● Use Default (disabled) for Auto-Closing Events
○ Enable Auto-Close Events
○ Disable Auto-Close Events

Auto-Close Note

The Schedule dialog box lets you 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 2.8.5, “Manage Notification Groups”.

- **Use SNMP Traps**: Indicates that an SNMP trap should be raised for this rule. For details on the SNMP configuration within MySQL Enterprise Monitor, see SNMP Trap configuration.

- **Auto-Close**: Specifies whether the rule should use the auto-close system. Events triggered by rules can be automatically closed when the situation that triggered the rule is no longer in effect. Some rules are configured with auto-close enabled by default. To determine which advisors this applies to, see Section B.9, “Advisor/Graph Reference”. For more information on closing and auto-closing events, see Section 2.4.1, “Closing an Event”.

  You can configure individual rules to either use the default auto-close settings for the advisor, to enable auto-close events, or to disable auto-close events. If auto-close of events has been enabled, you can specify the text that will be added to the event closure by adding it to the **Auto-Close Note** textbox.

Once you have set the schedule parameters, click the SCHEDULE button to schedule the advisor.

You can schedule multiple rules by selecting the checkbox next to each rule and clicking the SCHEDULE button is an effective way to schedule multiple rules.

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 5.3.1, “Creating a MySQL User Account for the Monitor Agent”.

2.3.2.1. 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.

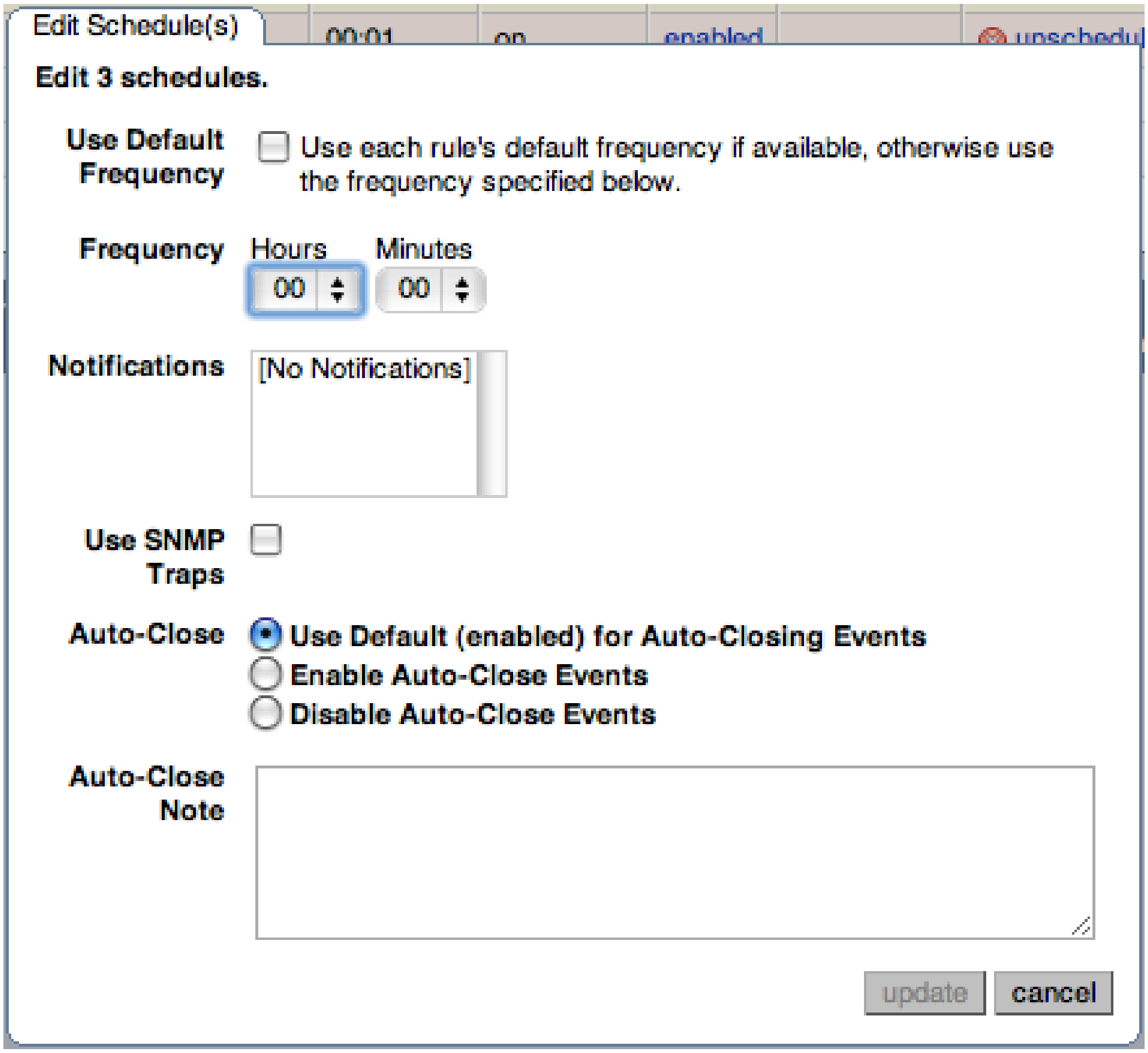
The Advanced tab gives detailed information about how this rule is implemented.

2.3.3. 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 2.5. MySQL Enterprise Dashboard: Editing Rules



Edit Schedule(s) 00:01 on enabled unschedul

Edit 3 schedules.

Use Default Frequency □ Use each rule's default frequency if available, otherwise use the frequency specified below.

Frequency Hours 00 Minutes 00

Notifications [No Notifications]

Use SNMP Traps □

Auto-Close
● Use Default (enabled) for Auto-Closing Events
○ Enable Auto-Close Events
○ Disable Auto-Close Events

Auto-Close Note

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 D, *Data Collection Items*.

In Figure 2.5, “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.

2.4. The EVENTS Tab

> **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.

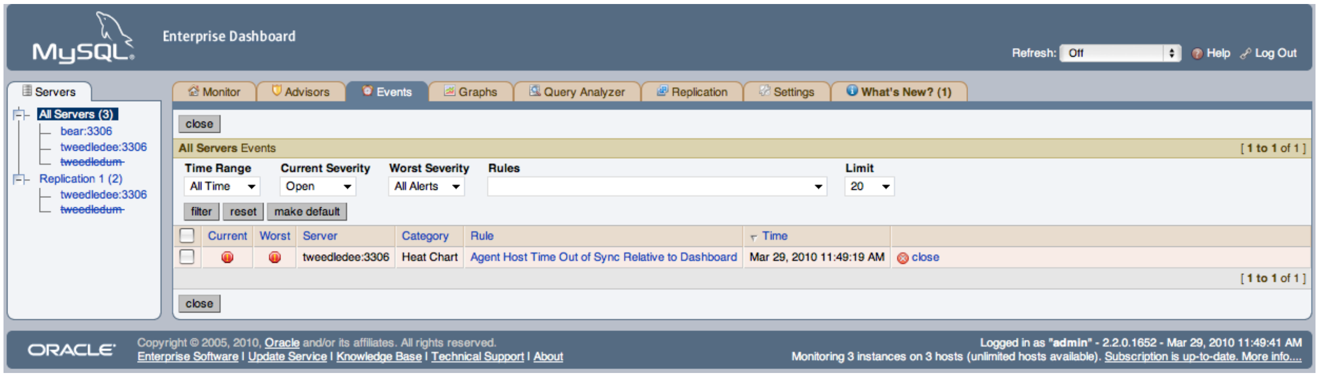
For convenience, the event states are also represented by icons, as defined in the table below. The table also shows the relative level of the alert from the highest (Critical) to the lowest (unknown). The order represented here is used when identifying when an event has transitioned between levels (for 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 is an informational alert.
✓	A green tick 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 2.8.5, “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 2.6. MySQL Enterprise Dashboard: Events Screen



All Servers Events [1 to 1 of 1]

Time Range: All Time
Current Severity: Open
Worst Severity: All Alerts
Rules:
Limit: 20

□	Current	Worst	Server	Category	Rule	Time	
□	🛑	🛑	tweedledee:3306	Heat Chart	Agent Host Time Out of Sync Relative to Dashboard	Mar 29, 2010 11:49:19 AM	⊗ close

[1 to 1 of 1]

Copyright © 2005, 2010, Oracle and/or its affiliates. All rights reserved.
Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "admin" - 2.2.0.1652 - Mar 29, 2010 11:49:41 AM
Monitoring 3 instances on 3 hosts (unlimited hosts available). Subscription is up-to-date. More info....

The event table has the following columns:

- **Current**: An icon indicating the current severity of the alert
- **Worst**: An icon indicating the worst severity recorded for this alert
- **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 green tick 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 2.6, “MySQL Enterprise Dashboard: Events Screen”.

The server shown in Figure 2.6, “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.

2.4.1. Closing an Event

The creation of an event by an advisor rule signifies that there an issue has been raised which may require closer attention. Once the event has been identified, you should investigate the issue that triggered the event, rectify the issue or problem (or choose to ignore it), and then close the event when you are satisfied that the event is not having a significant impact on your server or servers.

Some of the advisors are designed to identify issues in your server or MySQL Enterprise Monitor environment that may either be of a transient nature, or that may indicate a temporary issue (such as a replication slave being unavailable). For these advisors, you can schedule events to automatically be closed when the event moves from notification status back to the `Ok` state.

You can configure auto-close functionality on all advisors through the **Advisors** tab when you schedule the advisor against one or more servers. When auto-close is enabled, the event will remain open while the condition that triggered the event is still in effect. When the condition is no longer broken, the event will automatically be closed with the configured auto-close test. You can still manually close events that have have been configured with auto-close enabled. For more information on scheduling rules, see Section 2.3.2, “Scheduling Rules”.

To close an individual alert click the **CLOSE** link in the **OPERATIONS/NOTES** column. Document the resolution using the **RESOLUTION NOTES** text area and choose the CLOSE button. During the closing operation, you can also reconfigure the rule scheduling that triggered this event, and the auto-close settings. For more information on configuring rule scheduling and auto closing, see Section 2.3.2, “Scheduling Rules”.

To close a number of alerts simultaneously, select the checkbox beside each alert that you wish to close and then click the CLOSE button.

When closing individual or multiple events, a notification window will be shown indicating what operations have been completed. The events will remain in the displayed event list, but the **CLOSE** link will instead be a link to the `resolution notes`. You can update the active list by clicking FILTER to re-filter the event display.

An historical list of all events, including closed events, is available by using the **CURRENT SEVERITY** and setting it to `Closed`. The list will show all of the closed events for a given time range and the servers selected in the server tree.

2.4.2. Notification of Events

Notification of an event takes place when a rule is executed and the severity level for a given rule changes from a lower status to a higher one. This occurs on the first change to each individual severity level until the event has been closed.

For example, the table shows the severity levels raised during the lifecycle of a single event, showing when a notification is actually raised.

From Level	To Level	Notification Raised?
`Ok`	`Info`	`Yes`
`Info`	`Warning`	`Yes`
`Warning`	`Info`	`No`
`Warning`	`Critical`	`Yes`
`Critical`	`Warning`	`No`

If auto-close has been enabled for a given rule, then the above notification sequence is still valid, but the effect of the auto-close system is that the event will automatically be closed when the event reaches the `Ok` state. This will re-enable notification when the advisor creates a new event.

SNMP notifications are sent each time a given rule is executed, irrespective of the current or returned state.

2.5. The GRAPHS Tab

> **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** tab 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** tab 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** tab. 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 2.2.1, “The Server Graphs and Critical Events”. As with the **MONITOR** tab, the data shown in the graphs is determined by the server or group of servers selected in the server tree.

The **GRAPHS** tab 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.

All graphs support the ability to drag and select a specific area of the graph. When you select an area of the graph, the display changes to the **QUERY ANALYZER** tab and displays the queries that were being executed during the selected time period. You can also zoom into the graph by selecting specific time periods. For more information on using the graphs to monitor the execution of specific queries, see Section 3.2.2, “Using Graphs to Identify Queries”.

2.5.1. Displaying Graphs

The total number of graphs varies depending upon your subscription level. The four graphs that appear by default on the **MONITOR** tab are:

- Hit Ratios
- Database Activity
- Connections
- CPU Utilization

When the **GRAPHS** tab 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** tab rather than on the **MONITOR** tab. Additionally, you can only show a maximum of six graphs on the **MONITOR** tab; the remaining graphs can only be viewed from the **GRAPHS** tab.

2.5.2. 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** tab but have *no* effect on the graphs on the **MONITOR** tab.

To change the graphs both here and on the **MONITOR** tab 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** tab as described in Section 2.2.1, “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.

2.5.3. 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 in order 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** tab. You cannot change the time span of the graphs that appear on the **MONITOR** tab. Changes apply only to the current user; defaults for other users will be unchanged.

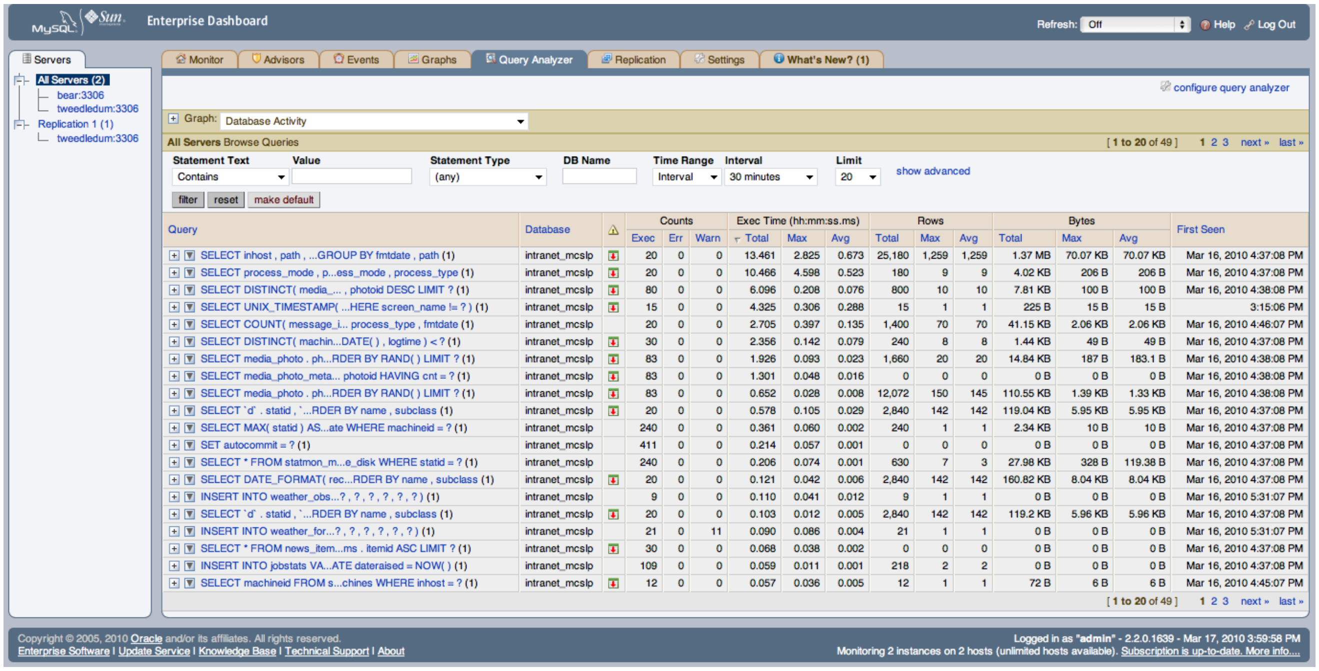
Use the RESET button to cancel your changes.

2.6. The QUERY ANALYZER Tab

Query Analyzer lets you 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.

Query Analyzer works by collecting information about SQL statements that your MySQL client application sends to the MySQL server, either by using the MySQL Enterprise Agent proxy service, or by using the MySQL Enterprise Monitor Plugin for a given connector that sends the information directly to MySQL Enterprise Service Manager.

Figure 2.7. MySQL Enterprise Dashboard: Query Analyzer



Graph: Database Activity

All Servers Browse Queries

Statement Text: Contains | Value: | Statement Type: (any) | DB Name: | Time Range: Interval | Interval: 30 minutes | Limit: 20

Query	Database	⚠	Counts			Exec Time (hh:mm:ss.ms)			Rows			Bytes			First Seen
			Exec	Err	Warn	Total	Max	Avg	Total	Max	Avg	Total	Max	Avg	
SELECT inhost , path , ...GROUP BY fmtdate , path (1)	intranet_mcslp	⬇	20	0	0	13.461	2.825	0.673	25,180	1,259	1,259	1.37 MB	70.07 KB	70.07 KB	Mar 16, 2010 4:37:08 PM
SELECT process_mode , p...ess_mode , process_type (1)	intranet_mcslp	⬇	20	0	0	10.466	4.598	0.523	180	9	9	4.02 KB	206 B	206 B	Mar 16, 2010 4:37:08 PM
SELECT DISTINCT(media_... , photoid DESC LIMIT ? (1)	intranet_mcslp	⬇	80	0	0	6.096	0.208	0.076	800	10	10	7.81 KB	100 B	100 B	Mar 16, 2010 4:38:08 PM
SELECT UNIX_TIMESTAMP(...HERE screen_name != ?) (1)	intranet_mcslp	⬇	15	0	0	4.325	0.306	0.288	15	1	1	225 B	15 B	15 B	3:15:06 PM
SELECT COUNT(message_i... process_type , fmtdate (1)	intranet_mcslp		20	0	0	2.705	0.397	0.135	1,400	70	70	41.15 KB	2.06 KB	2.06 KB	Mar 16, 2010 4:46:07 PM
SELECT DISTINCT(machin...DATE( ) , logtime) < ? (1)	intranet_mcslp	⬇	30	0	0	2.356	0.142	0.079	240	8	8	1.44 KB	49 B	49 B	Mar 16, 2010 4:37:08 PM
SELECT media_photo . ph...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	⬇	83	0	0	1.926	0.093	0.023	1,660	20	20	14.84 KB	187 B	183.1 B	Mar 16, 2010 4:38:08 PM
SELECT media_photo_meta... photoid HAVING cnt = ? (1)	intranet_mcslp	⬇	83	0	0	1.301	0.048	0.016	0	0	0	0 B	0 B	0 B	Mar 16, 2010 4:38:08 PM
SELECT media_photo . ph...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	⬇	83	0	0	0.652	0.028	0.008	12,072	150	145	110.55 KB	1.39 KB	1.33 KB	Mar 16, 2010 4:38:08 PM
SELECT `d` . statid , `...RDER BY name , subclass (1)	intranet_mcslp	⬇	20	0	0	0.578	0.105	0.029	2,840	142	142	119.04 KB	5.95 KB	5.95 KB	Mar 16, 2010 4:37:08 PM
SELECT MAX(statid) AS...ate WHERE machineid = ? (1)	intranet_mcslp		240	0	0	0.361	0.060	0.002	240	1	1	2.34 KB	10 B	10 B	Mar 16, 2010 4:37:08 PM
SET autocommit = ? (1)	intranet_mcslp		411	0	0	0.214	0.057	0.001	0	0	0	0 B	0 B	0 B	Mar 16, 2010 4:37:08 PM
SELECT * FROM statmon_m...e_disk WHERE statid = ? (1)	intranet_mcslp		240	0	0	0.206	0.074	0.001	630	7	3	27.98 KB	328 B	119.38 B	Mar 16, 2010 4:37:08 PM
SELECT DATE_FORMAT(rec...RDER BY name , subclass (1)	intranet_mcslp	⬇	20	0	0	0.121	0.042	0.006	2,840	142	142	160.82 KB	8.04 KB	8.04 KB	Mar 16, 2010 4:37:08 PM
INSERT INTO weather_obs...? , ? , ? , ? , ? , ?) (1)	intranet_mcslp		9	0	0	0.110	0.041	0.012	9	1	1	0 B	0 B	0 B	Mar 16, 2010 5:31:07 PM
SELECT `d` . statid , `...RDER BY name , subclass (1)	intranet_mcslp	⬇	20	0	0	0.103	0.012	0.005	2,840	142	142	119.2 KB	5.96 KB	5.96 KB	Mar 16, 2010 4:37:08 PM
INSERT INTO weather_for...? , ? , ? , ? , ? , ?) (1)	intranet_mcslp		21	0	11	0.090	0.086	0.004	21	1	1	0 B	0 B	0 B	Mar 16, 2010 5:31:07 PM
SELECT * FROM news_item...ms . itemid ASC LIMIT ? (1)	intranet_mcslp	⬇	30	0	0	0.068	0.038	0.002	0	0	0	0 B	0 B	0 B	Mar 16, 2010 4:37:08 PM
INSERT INTO jobstats VA...ATE dateraised = NOW( ) (1)	intranet_mcslp		109	0	0	0.059	0.011	0.001	218	2	2	0 B	0 B	0 B	Mar 16, 2010 4:37:08 PM
SELECT machineid FROM s...chines WHERE inhost = ? (1)	intranet_mcslp	⬇	12	0	0	0.057	0.036	0.005	12	1	1	72 B	6 B	6 B	Mar 16, 2010 4:45:07 PM

Copyright © 2005, 2010 Oracle and/or its affiliates. All rights reserved.
Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "admin" - 2.2.0.1639 - Mar 17, 2010 3:59:58 PM
Monitoring 2 instances on 2 hosts (unlimited hosts available). Subscription is up-to-date. More info....

For more information using the Query Analyzer, including how to feed and provide queries to MySQL Enterprise Service Manager, how to use the user interface, and how to drill down to find the queries and information that you need, see Chapter 3, *Using the Query Analyzer*.

2.7. The REPLICATION Tab

> **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.

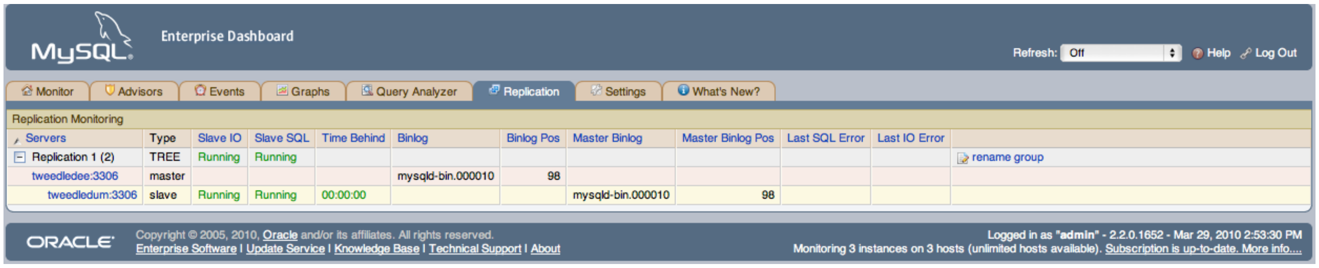
Navigate to the `Replication` page by choosing the **REPLICATION** tab. This page provides a quick summary view of the state of your replication servers or, if you wish, you can drill down and determine specifics about any master or slave.

> **Note**
>
> Servers, whether masters or slaves, must be monitored in order 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 automatically discovered and a grouping is created. This grouping shows up on the replication page and also in the **HEAT CHART** on the **MONITOR** tab. 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 2.8. MySQL Enterprise Dashboard: Replication Groups



MySQL
Enterprise Dashboard
Refresh: Off
Help
Log Out
Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New?

Replication Monitoring

Servers	Type	Slave IO	Slave SQL	Time Behind	Binlog	Binlog Pos	Master Binlog	Master Binlog Pos	Last SQL Error	Last IO Error	
⊟ Replication 1 (2)	TREE	Running	Running								rename group
tweedledee:3306	master				mysqld-bin.000010	98					
tweedledum:3306	slave	Running	Running	00:00:00			mysqld-bin.000010	98			

ORACLE
Copyright © 2005, 2010, Oracle and/or its affiliates. All rights reserved.
Enterprise Software | Update Service | Knowledge Base | Technical Support | About
Logged in as "admin" - 2.2.0.1652 - Mar 29, 2010 2:53:30 PM
Monitoring 3 instances on 3 hosts (unlimited hosts available). Subscription is up-to-date. More info....

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 in order 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 non-slaves to a replication grouping. For more information about server groupings see Section 2.8.3.2, “Grouping Servers”.

2.7.1. 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** tab.

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 IO 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 IO 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.

2.8. The Settings Tab

> **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 initial installation lets you configure your MySQL Enterprise credentials and outgoing email settings. This section explores the configuration settings in more detail, and 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, which is explained in Section 2.3, “The **Advisors** Tab”.

To get to the **Settings** tab, open the Dashboard and choose the **Settings** tab.

2.8.1. Global Settings

The **Global Settings** control the main configuration parameters for the entire MySQL Enterprise Monitor system, including your email notifications, data purge, and Enterprise website credentials.

The **Global Settings** tab is divided into these sections:

- **Outgoing Email Settings**

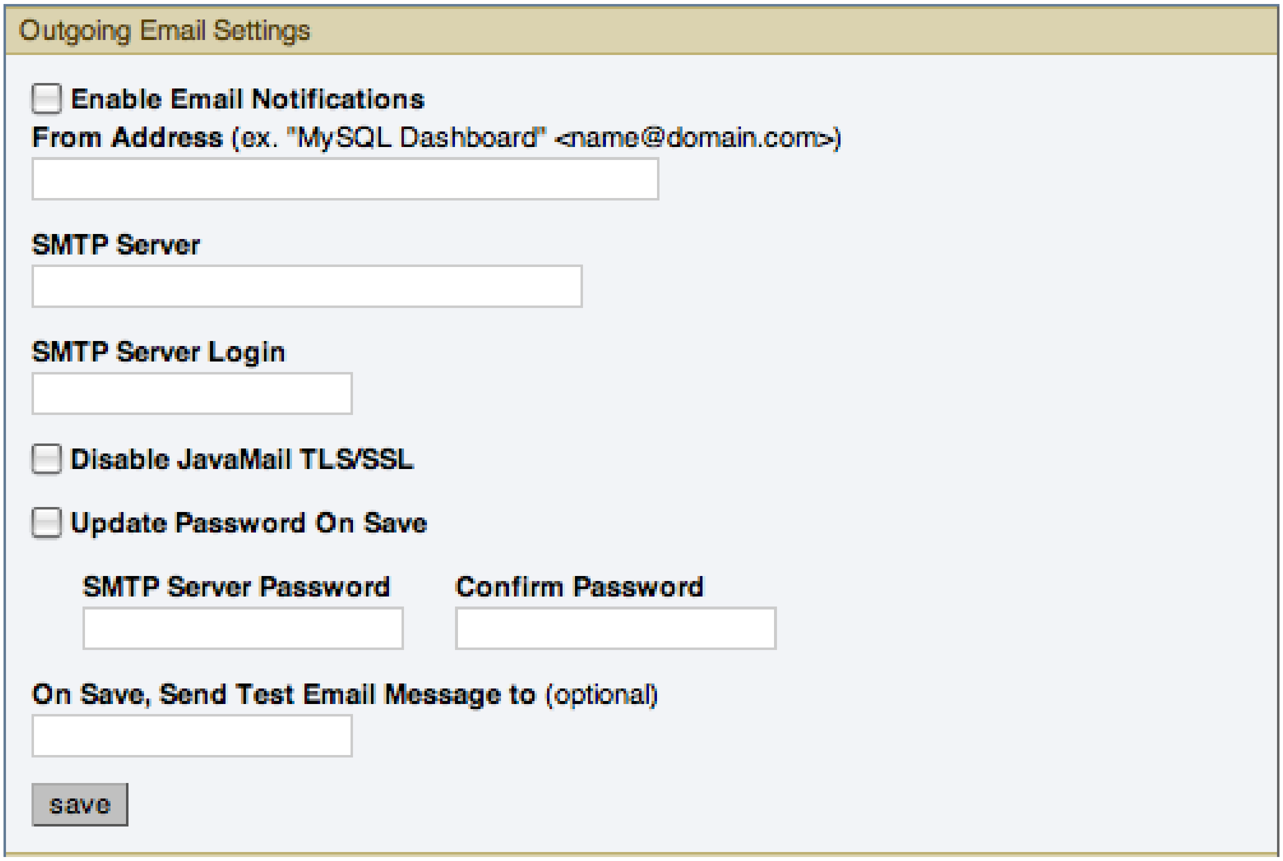
  Configures the settings for email notifications by MySQL Enterprise Service Manager. Configure the **From Address SMTP Server** settings. If your server requires authorization, complete the necessary server login details, and whether SSL is required.

  Alert notification via 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 configure the email settings that apply to the currently logged-in user.

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

  Figure 2.9. MySQL Enterprise Dashboard Settings: Outgoing Email



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, specify the port, separated from the server name by 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.

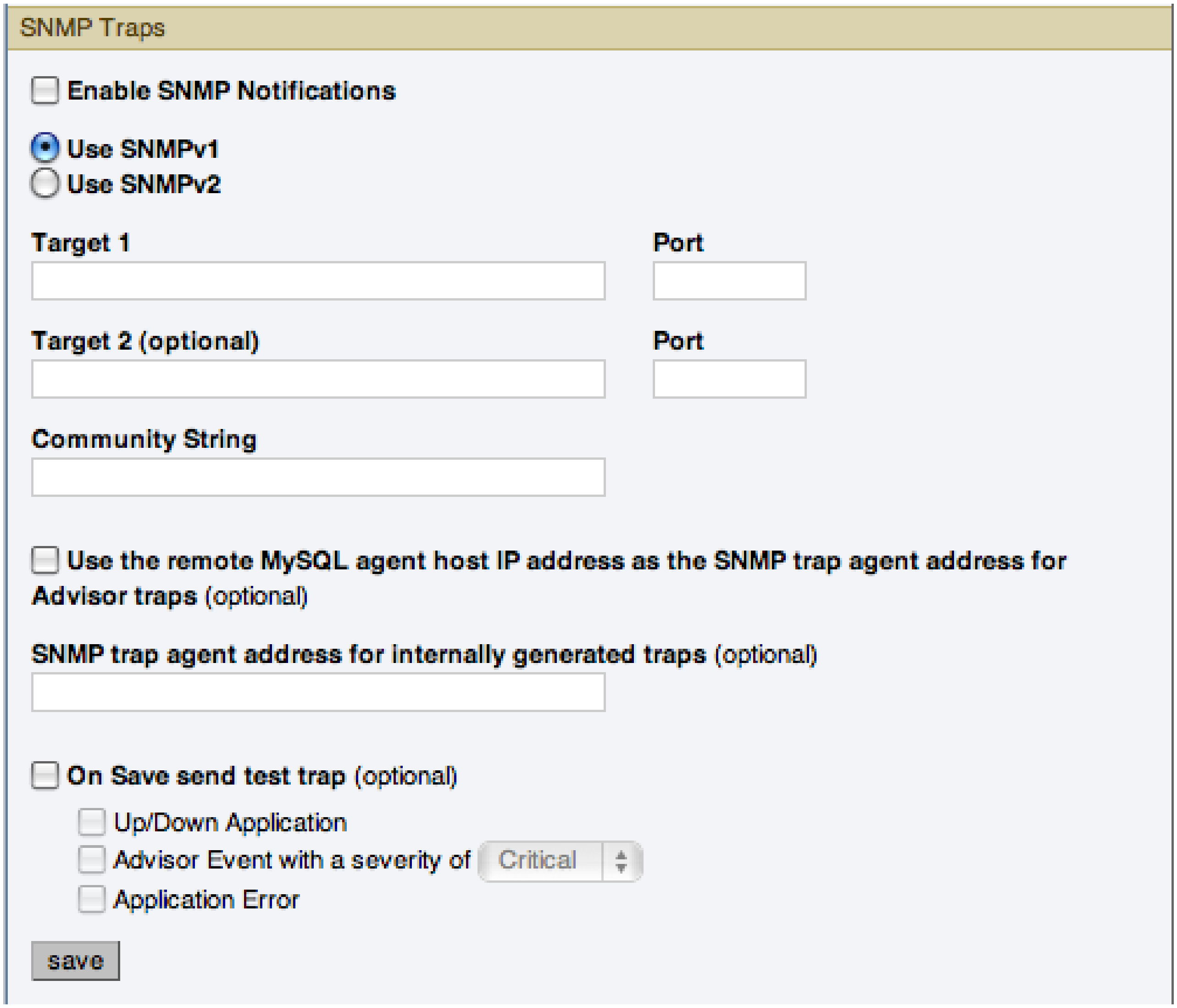
To test your configuration immediately, add an email address to the **On Save, Send Test Email Message to** box.

For more information about **Outgoing Email Settings**.

- **SNMP Traps**

  The **SNMP Traps** section of the **Global Settings** tab lets you 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. To enable them, click the **Enable SNMP Notifications** checkbox.

Figure 2.10. MySQL Enterprise Dashboard Settings: SNMP Traps



SNMP Traps

□ Enable SNMP Notifications

● Use SNMPv1
○ Use SNMPv2

Target 1 ______ Port ______

Target 2 (optional) ______ Port ______

Community String ______

□ Use the remote MySQL agent host IP address as the SNMP trap agent address for Advisor traps (optional)

SNMP trap agent address for internally generated traps (optional) ______

□ On Save send test trap (optional)

- □ Up/Down Application
- □ Advisor Event with a severity of Critical
- □ Application Error

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 2.3.2, “Scheduling Rules”.

The Management Information Base (MIB) file associated with SNMP trapping is called `MONITOR.MIB`. For the location this file see Section C.1.4, “The Management Information Base (MIB) File”.

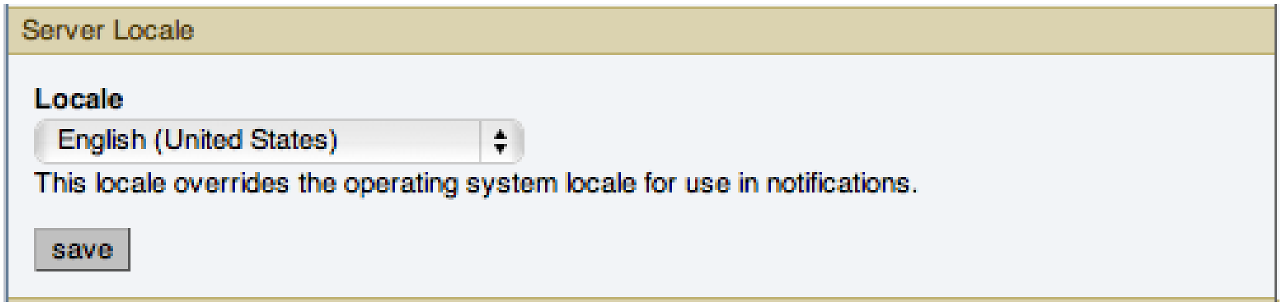
> **Note**
>
> The MIB file supplied with MySQL Enterprise Service Manager is not directly compatible with OpenNMS. For a sample of the modifications that need to be made, please see the comments for Bug#41714.

- **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

Figure 2.11. MySQL Enterprise Dashboard Settings: Server Locale



Server Locale

Locale

English (United States)

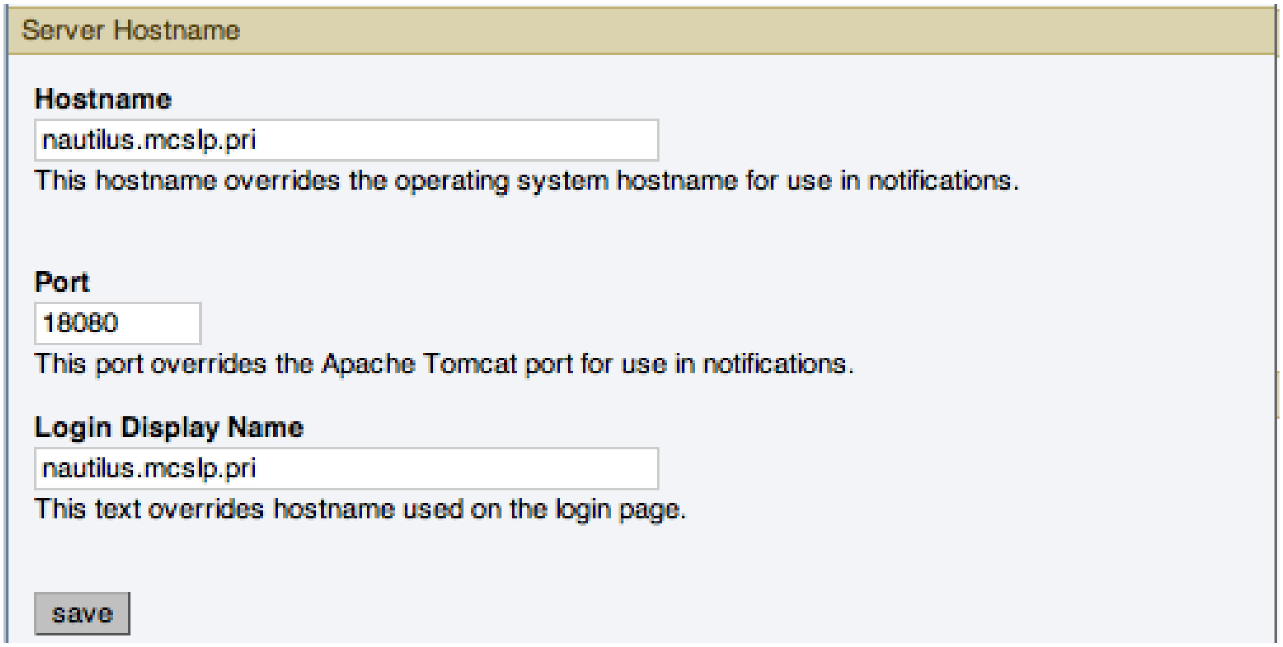
This locale overrides the operating system locale for use in notifications.

The initial value in this drop down list box is the locale for the OS on which the Dashboard is running.

- **SERVER HOSTNAME**

  You can alter the hostname and port used to identify the MySQL Enterprise Service Manager when reporting notifications. This can be useful if the server hostname, and the hostname you want to use when connecting to your MySQL Enterprise Service Manager

Figure 2.12. MySQL Enterprise Dashboard Settings: Server Hostname



Server Hostname

Hostname

nautilus.mcslp.pri

This hostname overrides the operating system hostname for use in notifications.

Port

18080

This port overrides the Apache Tomcat port for use in notifications.

Login Display Name

nautilus.mcslp.pri

This text overrides hostname used on the login page.

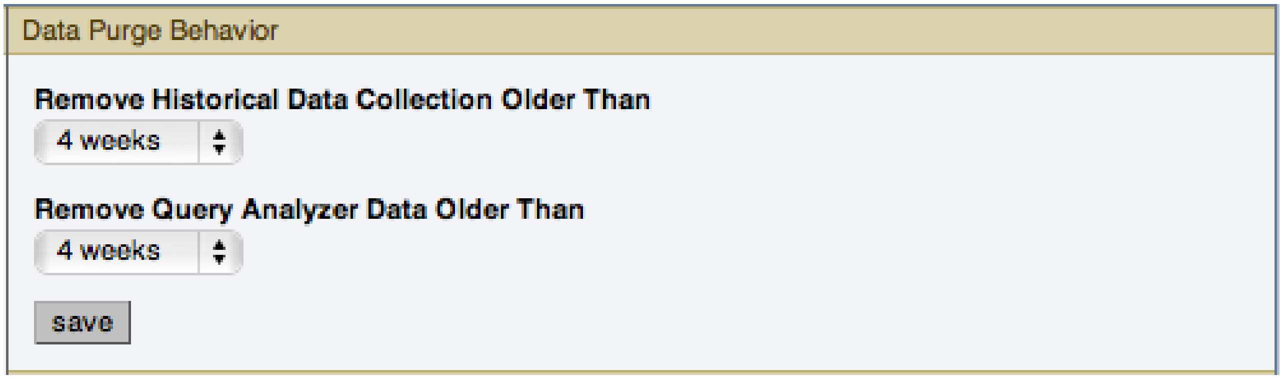
Note

Only change the **Port** setting if you have altered or redirected the original port used when installing MySQL Enterprise Service Manager. Entering the incorrect information does not affect the accessibility of your system, except when clicking links within the notification messages.

- **Data Purge Behavior**

  The **Data Purge Behavior** section of the **Global Settings** tab lets you remove old log files and also old data from the repository. The default purge interval is `28 days`. If you wish to purge data, change this setting by choosing from the drop-down list. Choosing `52 weeks`, for example, removes all data that is older than a year.

Figure 2.13. MySQL Enterprise Dashboard Settings: Data Purge



Data Purge Behavior

Remove Historical Data Collection Older Than

4 weeks

Remove Query Analyzer Data Older Than

4 weeks

Warning

Purging data permanently removes information from the repository. Since events are derived from data contained in the repository, they are 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 Query Analyzer Data Older Than** configures the duration that the query analyzer statistics and information about individual queries is retained.

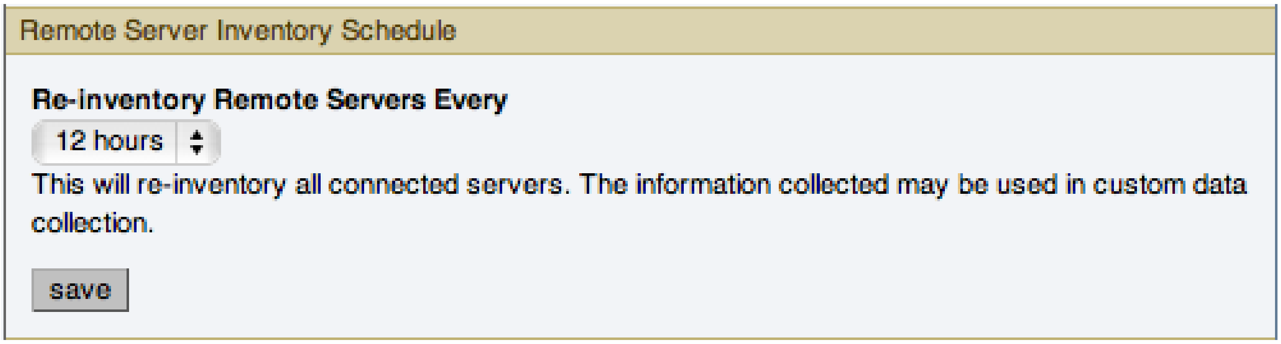
Notes for setting purge behavior:

- 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 is not 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.
- You cannot set the Query Analyzer purge period to a value greater than historical data collection purge period. Attempting to do so produces a warning: `An invalid Query Analyzer data life span was given. Value may not be larger than data life span.`.

- 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.

Figure 2.14. MySQL Enterprise Dashboard Settings: Remote Server Inventory Schedule



Remote Server Inventory Schedule

Re-Inventory Remote Servers Every

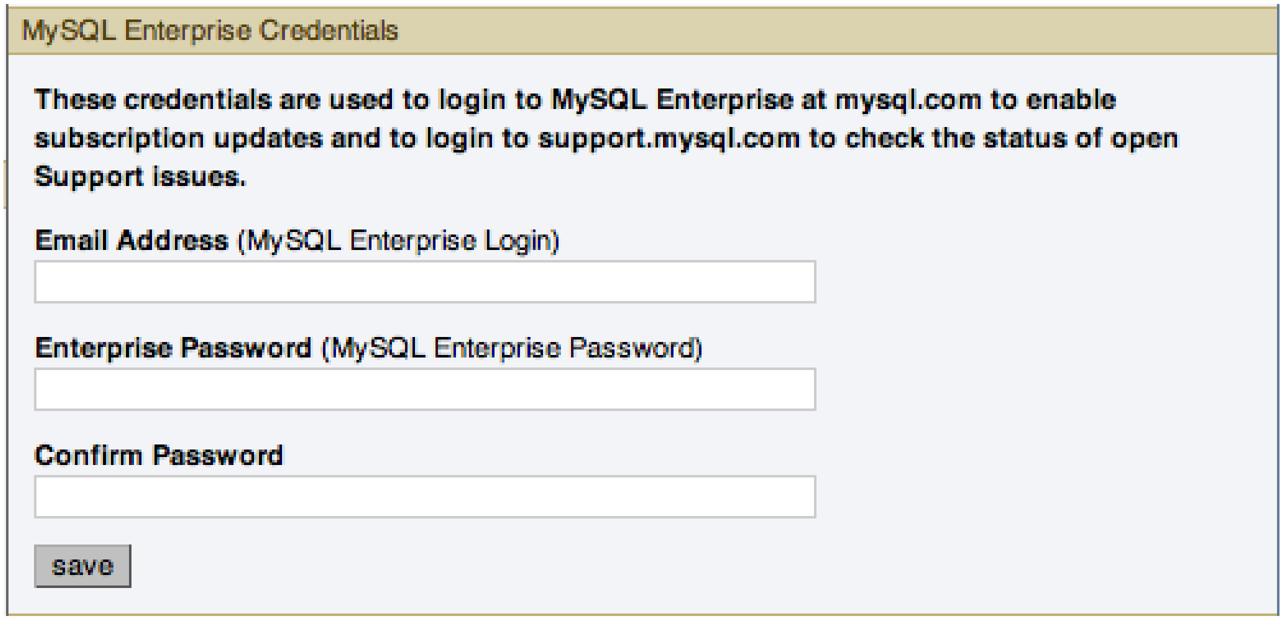
12 hours

This will re-inventory all connected servers. The information collected may be used in custom data collection.

- 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.

Figure 2.15. MySQL Enterprise Dashboard Settings: MySQL Enterprise Credentials



MySQL Enterprise Credentials

These credentials are used to login to MySQL Enterprise at mysql.com to enable subscription updates and to login to support.mysql.com to check the status of open Support issues.

Email Address (MySQL Enterprise Login)

Enterprise Password (MySQL Enterprise Password)

Confirm Password

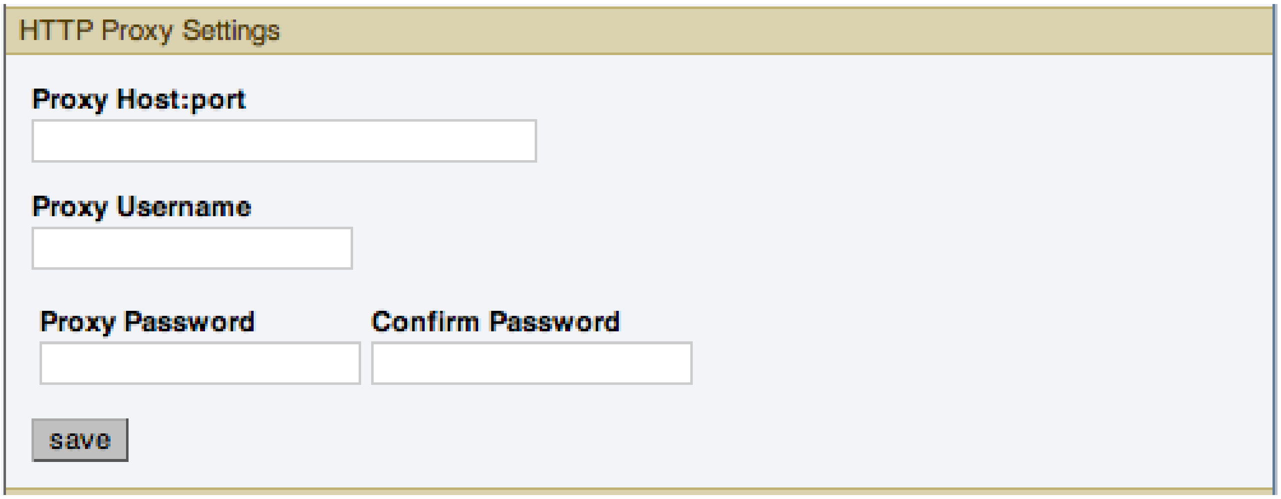
Note

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 2.8.4, "Managing Users". Specifying incorrect credentials results in the error message, "Your credentials do not appear to be valid."

- **HTTP Proxy Settings**

  You may want to update your HTTP Proxy Settings if your MySQL Enterprise Service Manager is not directly connected to the internet. The proxy settings are used when updating the information within the **What's New** tab and when accessing the MySQL Enterprise Website for authentication of your license and product key information. For more information, see Section 2.9, "The **What's New** Tab".

Figure 2.16. MySQL Enterprise Dashboard Settings: HTTP Proxy Settings



HTTP Proxy Settings

Proxy Host:port

Proxy Username

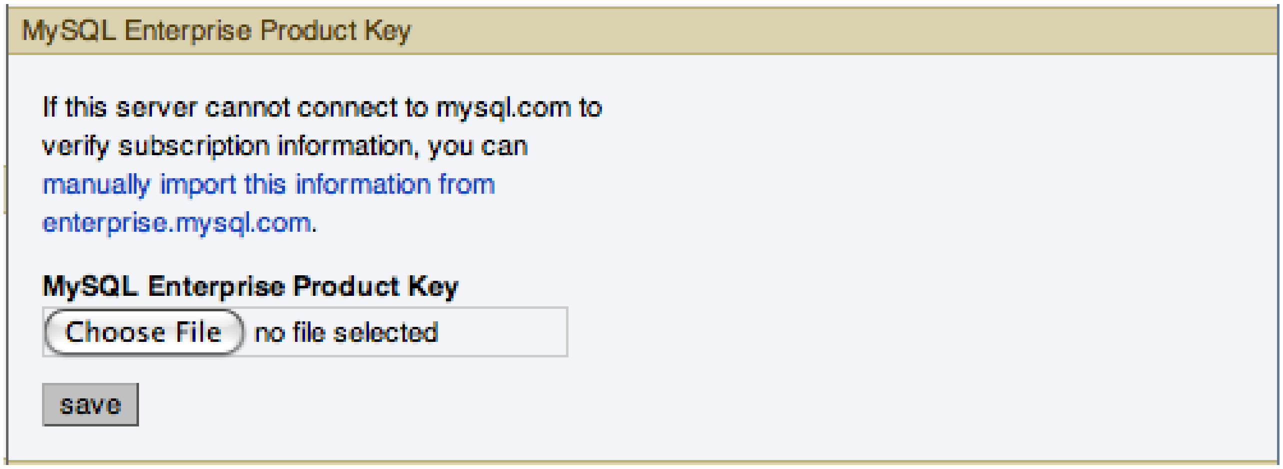
Proxy Password

Confirm Password

- **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.

Figure 2.17. MySQL Enterprise Dashboard Settings: Product Key



MySQL Enterprise Product Key

If this server cannot connect to mysql.com to verify subscription information, you can manually import this information from enterprise.mysql.com.

MySQL Enterprise Product Key

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.

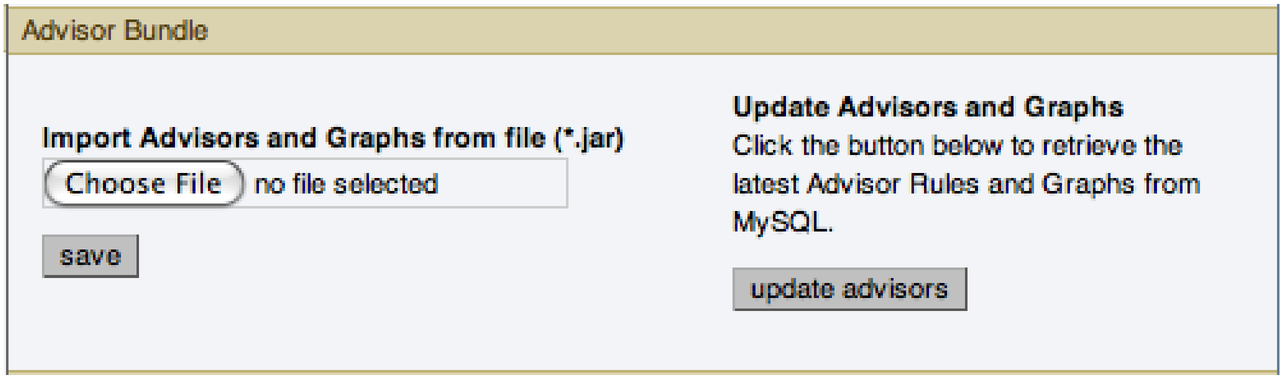
If you wish to switch from using your MySQL Enterprise credentials to using a product key to update MySQL Enterprise Monitor, you must first clear your credentials. Do this by removing the email address from the **MYSQL ENTERPRISE CREDENTIALS** section and then clicking the SAVE button. You may then 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 2.8.4, “Managing Users”. Specifying incorrect credentials results in the error message, “Your credentials do not appear to be valid.”

- ADVISOR BUNDLE

  You can update your Advisor Bundle and the associated rules and graphs by importing a new Advisor Bundle. To update your Advisor Bundle using a bundle file that you have previous downloaded:

Figure 2.18. MySQL Enterprise Dashboard Settings: Advisor Bundle



Advisor Bundle

Import Advisors and Graphs from file (*.jar)

Update Advisors and Graphs

Click the button below to retrieve the latest Advisor Rules and Graphs from MySQL.

1. Click CHOOSE FILE and locate your Advisor Bundle on your computer.

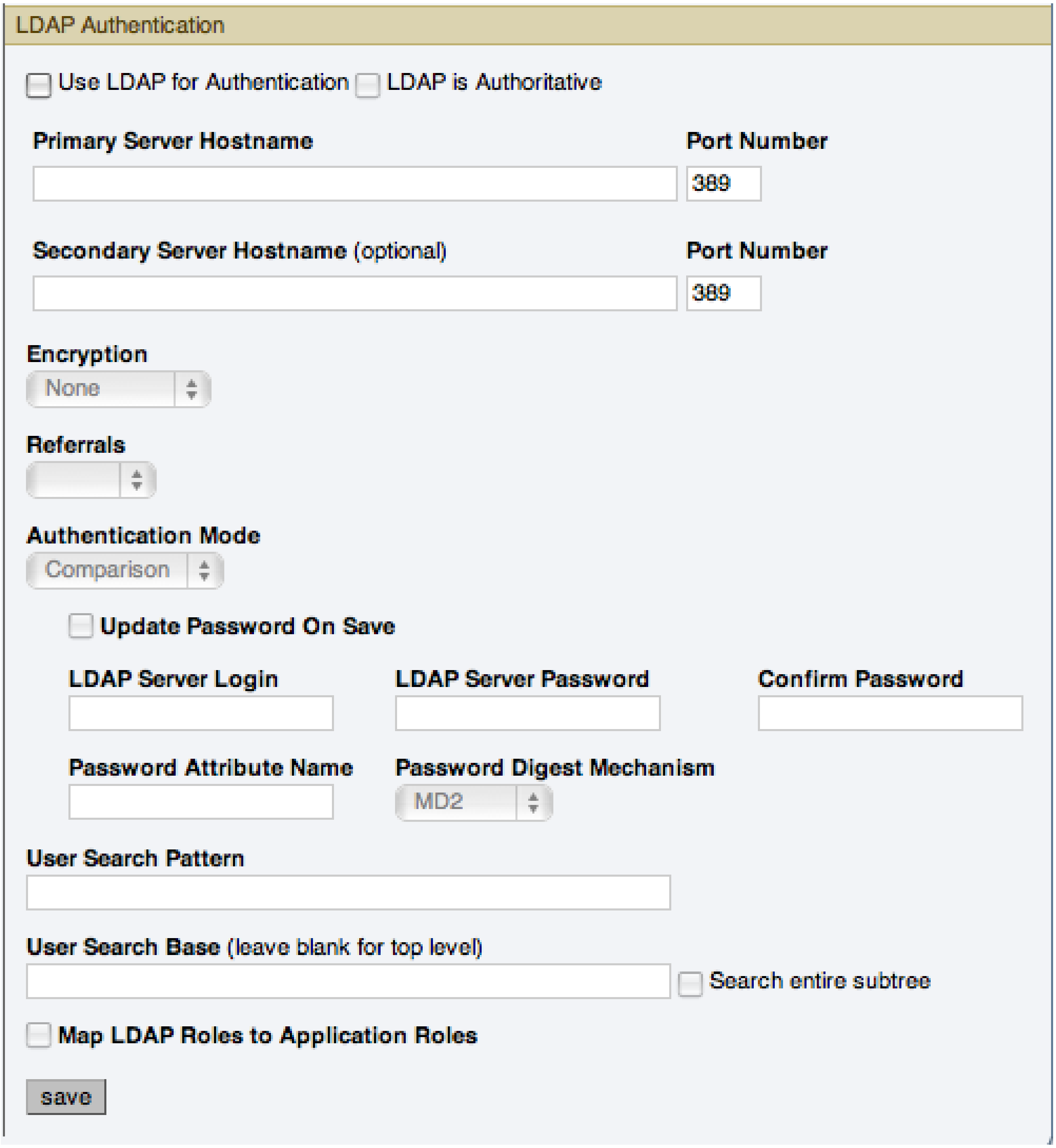
2. Click SAVE to upload the updated bundle and validate the contents.

To update the Advisor Bundle automatically from the MySQL Enterprise Website using your existing MySQL Enterprise Credentials, click the UPDATE ADVISORS button.

- **LDAP AUTHENTICATION**

  You can configure LDAP Authentication to be used for the users that are provided access to the MySQL Enterprise Dashboard. To use LDAP authentication it must have been enabled and configured within the settings.

Figure 2.19. MySQL Enterprise Dashboard Settings: LDAP Authentication



LDAP Authentication

□ Use LDAP for Authentication □ LDAP is Authoritative

Primary Server Hostname | Port Number: 389

Secondary Server Hostname (optional) | Port Number: 389

Encryption: None

Referrals:

Authentication Mode: Comparison

□ Update Password On Save

LDAP Server Login | LDAP Server Password | Confirm Password

Password Attribute Name | Password Digest Mechanism: MD2

User Search Pattern

User Search Base (leave blank for top level) □ Search entire subtree

□ Map LDAP Roles to Application Roles

The configurable elements for LDAP authentication are:

- **Use LDAP for Authentication**

  To enable LDAP authentication, click the **Use LDAP for Authentication** checkbox.

- **LDAP is Authoritative**

  If you want to make LDAP the authoritative (only) authentication mechanism, check the **LDAP is Authoritative** checkbox. Note that if you select this option and the LDAP service is misconfigured, you can lock yourself out of MySQL Enterprise Dashboard entirely.

- **Primary Server Hostname**

  Hostname or IP address of the primary LDAP directory server.

- **Port Number**

  Port number of the primary LDAP server. You must change this option to the port used for SSL connections if you have enabled encryption.

- **Secondary Server Hostname (optional)**

  Hostname or IP address of the secondary/failover LDAP directory server.

- **Port Number**

  Port number of the secondary/failover LDAP server. You must change this option to the port used for SSL connections if you have enabled encryption.

- **Encryption**

  Encryption type required for communication with the LDAP server(s). Supported options are `None`, `StartTLS`, and `SSL`.

- **Referrals**

  Authentication should follow any referrals provided by the server. The default is to use whatever the LDAP directory server is configured to do. If you are using Microsoft Windows Active Directory, you must set this option to `Follow`.

- **Authentication Mode**

  The authentication mode to use. Choices are `Bind as User`, which binds to the LDAP directory using the credentials given to login to MySQL Enterprise Service Manager. `Comparison` requires an LDAP login/password that can see the configured password attribute to make a comparison with the given credentials.

- **User Search Pattern**

  Pattern specifying the LDAP search filter to use after substitution of the username, where `{0}` marks where the username should be substituted for the DN.

- **User Search Base (leave blank for top level)**

  The entry to use as the base of the subtree containing users. If not specified, the search base is the top-level context.

- **Search entire subtree**

  The search scope. Set to `true` if you wish to search the entire subtree rooted at the *User Search Base entry. The default value of `false` requests a single-level search including only the top level.

- **Map LDAP Roles to Application Roles**

  Specifies whether MySQL Enterprise Service Manager should use the roles defined in LDAP to map to MySQL Enterprise Monitor application roles. If enabled, and LDAP is not configured to be authoritative, if a user authenticates successfully via LDAP and has a valid mapped role, they are granted permissions to the application. Roles are mapped according to the entries in the Application Role/LDAP Role(s) fields, which take comma-separated lists of LDAP roles to map to the given MySQL Enterprise Monitor roles.

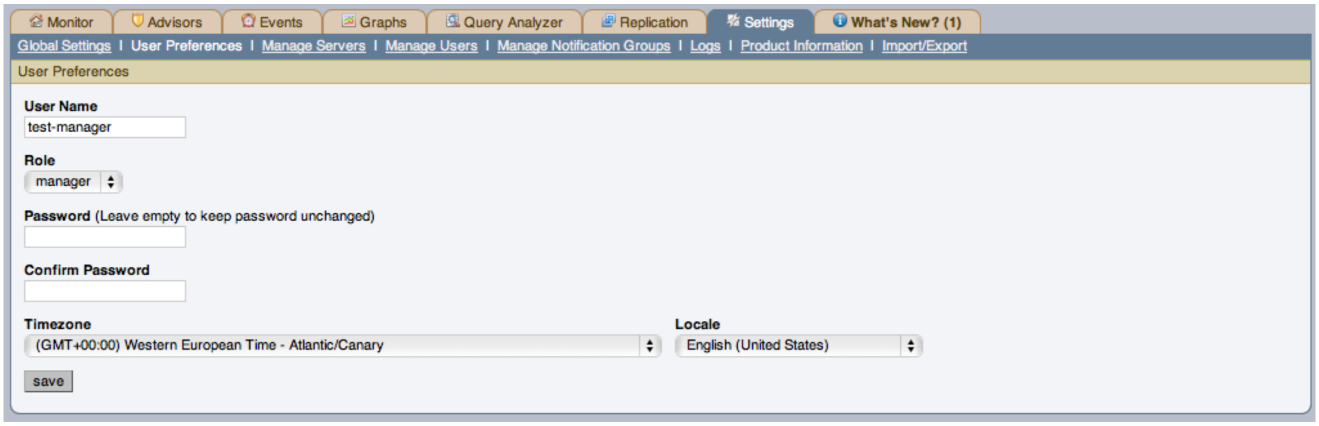
  If you select this option, you are provided with additional fields that let you configure how roles are looked up within the LDAP server.

For more information on LDAP authentication and integration with Tomcat, see Tomcat Documentation.

2.8.2. User Preferences

On this tab users can change their passwords, user name, and locale information.

Figure 2.20. MySQL Enterprise Dashboard: User Preferences


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New? (1)
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information | Import/Export
User Preferences

User Name
test-manager

Role
manager

Password (Leave empty to keep password unchanged)

Confirm Password

Timezone
(GMT+00:00) Western European Time - Atlantic/Canary

Locale
English (United States)

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 tab. The settings on this tab apply only to the user who is currently logged in.

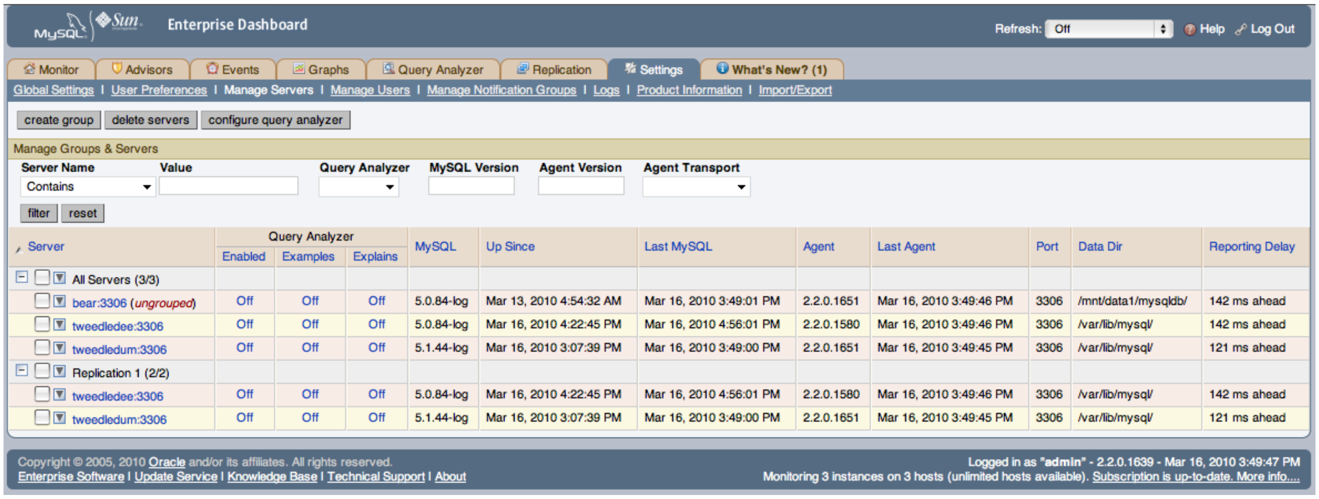
The MySQL Enterprise Service Manager determines the default value for the locale by looking at your browser settings. Changing this value, determines the language setting for any future logins to the Dashboard, overriding your browser settings.

> **Note**
>
> Be sure to set the correct time zone so that alerts are time stamped correctly.
>
> This setting applies only to the specific user.

2.8.3. Manage Servers

You can obtain a list of the currently monitored servers connected to the MySQL Enterprise Service Manager, and details of the current version and key configuration by using the **Manage Servers** tab.

Figure 2.21. MySQL Enterprise Dashboard: Manage Servers



Server Name: Contains | Value | Query Analyzer | MySQL Version | Agent Version | Agent Transport

Server	Query Analyzer			MySQL	Up Since	Last MySQL	Agent	Last Agent	Port	Data Dir	Reporting Delay
	Enabled	Examples	Explains								
All Servers (3/3)											
bear:3306 (*ungrouped*)	Off	Off	Off	5.0.84-log	Mar 13, 2010 4:54:32 AM	Mar 16, 2010 3:49:01 PM	2.2.0.1651	Mar 16, 2010 3:49:46 PM	3306	/mnt/data1/mysqldb/	142 ms ahead
tweedledee:3306	Off	Off	Off	5.0.84-log	Mar 16, 2010 4:22:45 PM	Mar 16, 2010 4:56:01 PM	2.2.0.1580	Mar 16, 2010 3:49:46 PM	3306	/var/lib/mysql/	142 ms ahead
tweedledum:3306	Off	Off	Off	5.1.44-log	Mar 16, 2010 3:07:39 PM	Mar 16, 2010 3:49:00 PM	2.2.0.1651	Mar 16, 2010 3:49:45 PM	3306	/var/lib/mysql/	121 ms ahead
Replication 1 (2/2)											
tweedledee:3306	Off	Off	Off	5.0.84-log	Mar 16, 2010 4:22:45 PM	Mar 16, 2010 4:56:01 PM	2.2.0.1580	Mar 16, 2010 3:49:46 PM	3306	/var/lib/mysql/	142 ms ahead
tweedledum:3306	Off	Off	Off	5.1.44-log	Mar 16, 2010 3:07:39 PM	Mar 16, 2010 3:49:00 PM	2.2.0.1651	Mar 16, 2010 3:49:45 PM	3306	/var/lib/mysql/	121 ms ahead

Copyright © 2005, 2010 Oracle and/or its affiliates. All rights reserved.
Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "admin" - 2.2.0.1639 - Mar 16, 2010 3:49:47 PM
Monitoring 3 instances on 3 hosts (unlimited hosts available). Subscription is up-to-date. More info....

The **Manage Servers** tab provides a list of servers, organized by groups, in a similar fashion to the main server navigation list. For each server, the following information is displayed:

- **Query Analyzer**

  Shows the Query Analyzer configuration for the server, showing whether Query Analyzer is enabled, whether the recovery of example queries is enabled, and whether explain plans for queries have been enabled.

  From this display, you can also control the Query Analyzer settings for each server by clicking the Off link for each server to change the settings. This brings up the **Configure Query Analyzer** pop-up window. For more information, see Section 3.2.3, “Query Analyzer Settings”.

- **MySQL**

  Shows the version number of the MySQL server being monitored.

- **Up Since**

  Shows the time when the MySQL server was started (the server up time).

- **Last MySQL**

  The last time communication with the MySQL server occurred.

- **Agent**

  Shows the version number of the MySQL Enterprise Agent server being monitored.

- **Last Agent**

  Shows the last time communication was received from the agent.

- **Port**

  The configured MySQL port number.

- **Data Dir**

  The data directory path for the MySQL server.

- **Reporting Delay**

  The time between when the data was obtained by the agent when it was received by MySQL Enterprise Service Manager.

You can search and filter the server list by using the fields and pop-ups at the top of the list. You can filter using one or more of following techniques:

- To filter by server name, select the match type (**Contains**, **Regex**, **Doesn't Contain**, or **Negative Regex**), and enter a string or regular expression into the **Value** field.

- To filter based on whether the Query Analyzer functionality has been enabled or disabled, use the **Query Analyzer** pop-up.

- To filter by MySQL server version, you can enter either an explicit version number into the **MySQL Version** field, or you can additional symbols to restrict the versions to different ranges. The field supports the following statements:

Statement	Example	Description
Version Prefix	`5.0`	Shows all MySQL servers matching the specified version. For example, `5.0` matches any version of MySQL from 5.0.0 to 5.0.99
Explicit Version	`5.0.84`, `=5.0.84`	Shows all MySQL servers matching the specific version number.
> Version	`> 5.0.84`	Shows all versions greater than the specified version.
>= Version	`>= 5.0.84`	Shows all versions greater than or equal to the specified version.

Statement	Example	Description
< Version	`< 5.0.84`	Shows all versions less than the specified version.
<= Version	`< 5.0.84`	Shows all versions less than or equal to the specified version.
String	`enterprise`, `log`	Shows all servers with the specified string in their version
! String or Version	`!enterprise`, `!5.0.84`	Shows all servers not having the specified string, or not matching the specified version.

You can combine multiple arguments to produce more complex formats. For example, to list all of the servers of 5.1 or higher that do not have enterprise in the version string, you could use `>=5.1 !enterprise`.

- **AGENT VERSION**

  Filters the list based on the MySQL Enterprise Agent version number. This can be useful if you want to know which means require an upgrade to a later version. The **AGENT VERSION** field supports the same selection criteria as the **MYSQL VERSION** field, allowing to select based on specific version numbers and strings.

- **AGENT TRANSPORT**

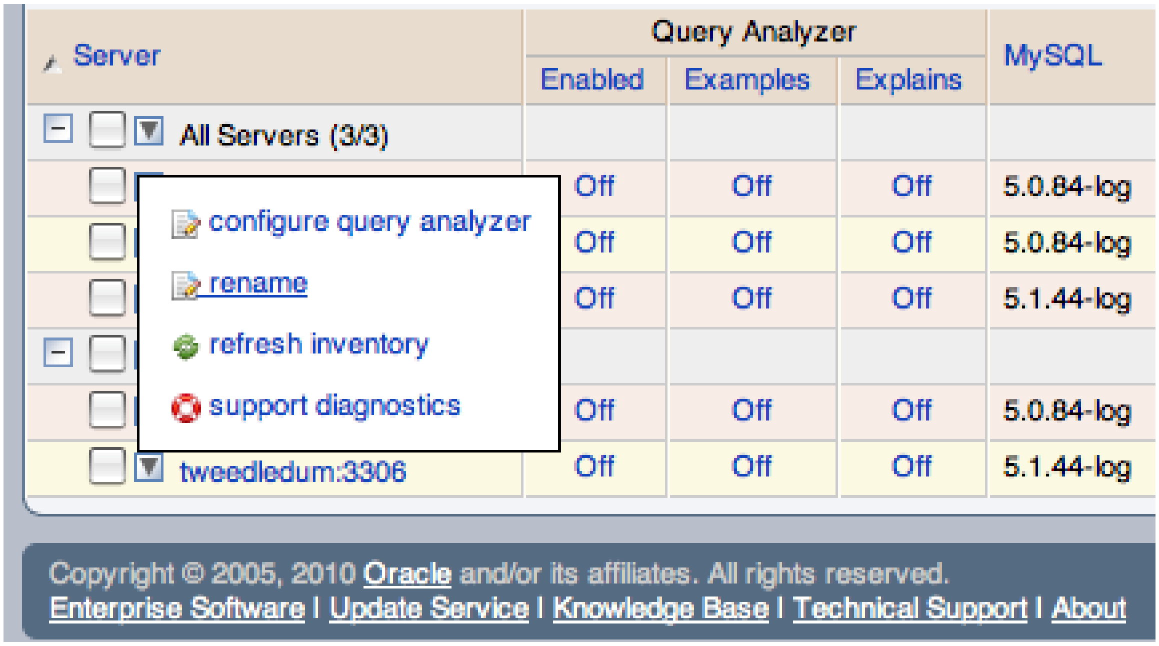
  Filters the list based on the protocol the Agent uses to communicate with the MySQL Enterprise Service Manager. You can select either standard HTTP or HTTPS/SSL-based connections.

After you make you selection, use the FILTER button to filter the server list. To reset your selections to show everything, click RESET.

Using the list of servers, you can create groups to more easily manage and monitor your servers according to the departments, applications or other groupings that you may want to use. For information on how to create and modify groups, see Section 2.8.3.2, “Grouping Servers”.

You can also perform simple administration on the server within the MySQL Enterprise Monitor framework, including configuring the Query Analyzer, renaming the server (as it appears within MySQL Enterprise Service Manager) or the group name, and updating the information about the server (re-inventory). All of these operations are handled by using hovering over the triangle next to the server name or group.

Figure 2.22. MySQL Enterprise Dashboard: Server/Group Pop-up



Server	Query Analyzer			MySQL
	Enabled	Examples	Explains	
⊟ □ ▼ All Servers (3/3)				
□	Off	Off	Off	5.0.84-log
□	Off	Off	Off	5.0.84-log
□	Off	Off	Off	5.1.44-log
⊟ □				
□	Off	Off	Off	5.0.84-log
□ ▼ tweedledum:3306	Off	Off	Off	5.1.44-log

Copyright © 2005, 2010 Oracle and/or its affiliates. All rights reserved.
Enterprise Software | Update Service | Knowledge Base | Technical Support | About

The pop-up provides a list of options, which are different depending on whether you have selected an individual server or a group. For the group, you can only rename the group, delete the group, or obtain a bundled version of the support diagnostics information. For individual servers, you can modify the Query Analyzer configuration, rename the server, refresh the server inventory, or obtain the support diagnostics information.

- To change the configuration of the Query Analyzer for the server, see Section 3.2.3, “Query Analyzer Settings”. This option is not available for a group.

- To rename a server choose RENAME, or to rename a group, choose RENAME GROUP. For information on renaming a server, see Section 2.8.3.1, “Renaming a Server”. To rename a group, you can change change within the pop-up prompt provided.

- To re-inventory your MySQL server and update version and configuration information, click the triangle and choose REFRESH INVENTORY. This triggers an update of the server information without affecting the schedule inventory process. You can use this when you have upgraded or made significant configuration changes to your server to update the inventory and ensure that the correct rules and advisors are being executed on your server.

- You can obtain the support diagnostics file for either individual servers, or entire server groups. To get a diagnostic report file, select SUPPORT DIAGNOSTICS from the pop-up menu. The information is provided as a Zip file that is downloaded to the machine. The information contained in the report includes detailed information about your server (or multiple servers if you selected a server group), including configuration, hardware, MySQL options/variables and historical graphs. To view the information extracted, unzip the downloaded file and double-click the `index.html`.

  For generic information about your MySQL Enterprise Monitor installation, see Diagnostic Report.

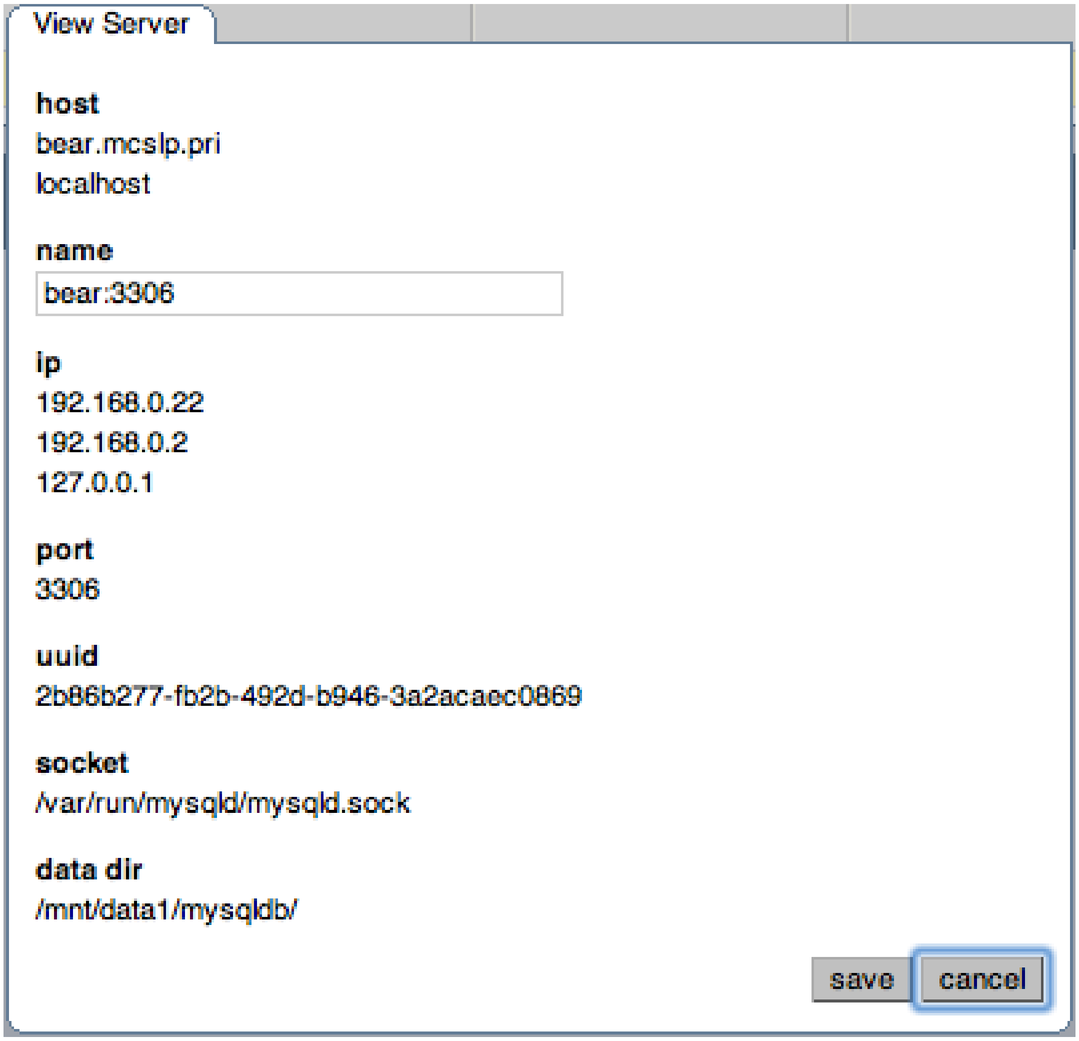
  > **Note**
  >
  > Renaming a server using the **MANAGE SERVERS** tab overrides all other settings, including changes made within the agent configuration.

2.8.3.1. Renaming a Server

You can rename an existing server without losing the current historical data or configuration information. Renaming the server also lets you make its name more descriptive according to the server's role within your organization. For example, you might 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. Enter 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.

Figure 2.23. MySQL Enterprise Dashboard: Server Renaming



host
bear.mcslp.pri
localhost

name
bear:3306

ip
192.168.0.22
192.168.0.2
127.0.0.1

port
3306

uuid
2b86b277-fb2b-492d-b946-3a2acaec0869

socket
/var/run/mysqld/mysqld.sock

data dir
/mnt/data1/mysqldb/

2.8.3.2. Grouping Servers

To help with server management, the Service Manager lets you group servers in any fashion you choose. For example, you can manage servers according to purpose. For example, you might group servers by whether the servers handle Internet or intranet data, by whether they power finance or HR applications, or by physical location.

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 is 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 Section 2.7, "The **REPLICATION** Tab". 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 2.8.3.3, "Removing a Server From the Dashboard".

Note

> Slaves removed from a replication group are rediscovered and re-added to that group.

To delete an group, you can use the **Delete Group** link. This deletes the group and removes the servers from 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 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.

2.8.3.3. 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 5.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 5.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 5.3.5.2, “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 no longer shows up in the Dashboard.

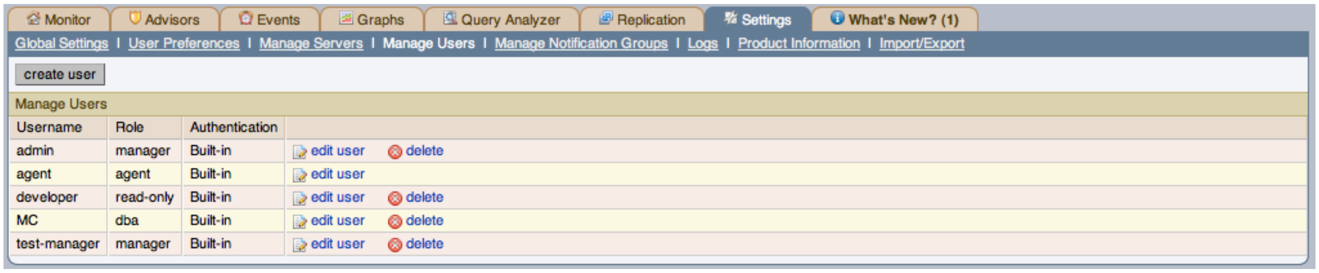
You can remove one or multiple servers at the same time. To remove a server or servers, click the checkbox next to each server that you want to delete, then click the DELETE SERVERS button. You are asked to confirm the deletion of the servers: click DELETE to delete the servers, or CANCEL to cancel the delete operation.

> **Note**
>
> You may remove a server from any group at any time. Removing the last server from a group also removes that group.

2.8.4. Managing Users

The **Manage Users** panel lets you create, delete and manage individual users that have access to MySQL Enterprise Service Manager

Figure 2.24. MySQL Enterprise Dashboard: Manage Users


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New? (1)
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information | Import/Export
create user

Manage Users

Username	Role	Authentication	
admin	manager	Built-in	edit user delete
agent	agent	Built-in	edit user
developer	read-only	Built-in	edit user delete
MC	dba	Built-in	edit user delete
test-manager	manager	Built-in	edit user delete

To log in to the Dashboard a user account is required. There are four types of users with varying privileges:

- **manager** role

  A user with the role of `manager` can perform all operations within MySQL Enterprise Service Manager, including changing the server configuration, adding and removing users, and viewing query analyzer and other data.

- **dba** role

  Users with the dba role should be database administrators who are using MySQL Enterprise Monitor to monitor servers and queries. They have a similar set of privileges to those with the manager, except that they cannot do the following:

  - Cannot add/remove or change users
  - Cannot change the MySQL Enterprise credentials used to communicate with the MySQL Enterprise website.
  - Cannot change the HTTP proxy settings.
  - Cannot update the Advisor bundle
  - Cannot change the LDAP authentication preferences.

  In addition to these limitations, you can control for each user with the dba role whether:

  - **View Query Analyzer tab**: Limits whether the user can see the Query Analyzer tab and its contents.
  - **View action (example) queries**: Limits whether the user can see the values within queries shown within the query analyzer.

  The setting for these abilities is provided when you select the dba role for a user.

- *read-only*

  The read-only role is identical to the dba role, but is unable to make any changes to the configuration of MySQL Enterprise Monitor, although they can view the current settings.

- **agent**

  The agent role is reserved for all users that you want to have access and ability to send agent information. You can use this if you want to provide individual logins for each agent.

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** tab. 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** tab. For more information, see Section 2.8.2, “User Preferences”.

If you installed the Advisors through the Dashboard you should have already configured the settings for the root role user. (See Section 2.8.1, “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 2.8.1, “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.

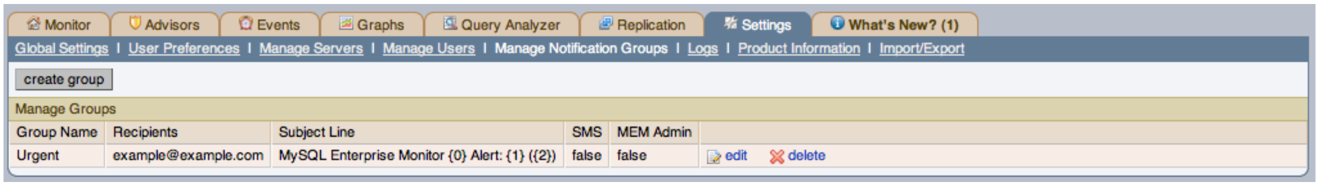
Users can be authenticated either using an internal authentication system, which stores the user, password and role information within MySQL Enterprise Monitor, or you can opt to use an external LDAP service. To use the LDAP service, you must still create each user, and then select the **Authenticate this user using LDAP** checkbox within the **Edit User** window. The username entered is authenticated against the configured LDAP server. For more information on configuring the LDAP server to be used, see **LDAP Authentication**.

To delete an existing user, click the delete link next to the user that you want to delete.

2.8.5. Manage Notification Groups

The **Manage Notification Groups** panels lets you create and manage the notification groups used when different notifications and warnings are distributed.

Figure 2.25. MySQL Enterprise Dashboard: Manage Notification Groups


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New? (1)
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information | Import/Export
create group
edit
delete

Manage Groups

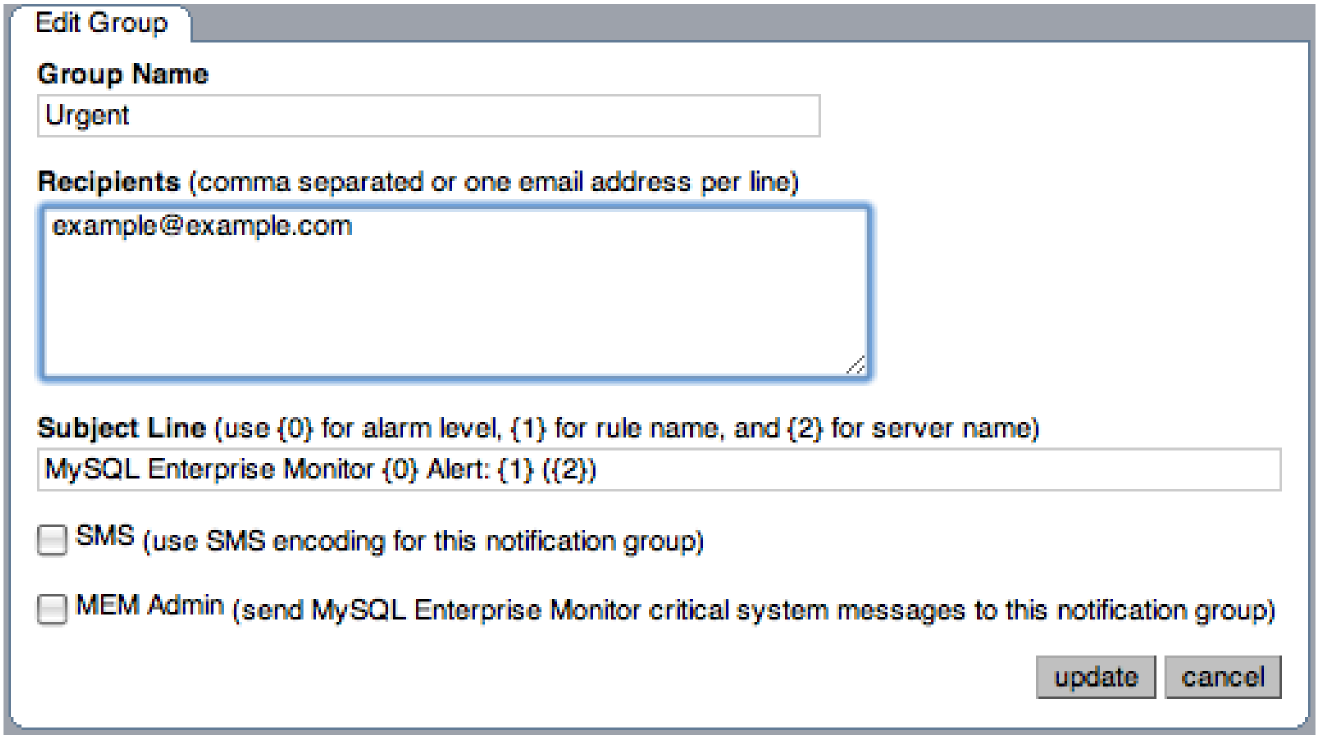
Group Name	Recipients	Subject Line	SMS	MEM Admin	
Urgent	example@example.com	MySQL Enterprise Monitor {0} Alert: {1} ({2})	false	false	edit delete

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 2.26. MySQL Enterprise Dashboard: Edit Notification Group



Group Name

Urgent

Recipients (comma separated or one email address per line)

example@example.com

Subject Line (use {0} for alarm level, {1} for rule name, and {2} for server name)

MySQL Enterprise Monitor {0} Alert: {1} ({2})

☐ SMS (use SMS encoding for this notification group)

☐ MEM Admin (send MySQL Enterprise Monitor critical system messages to this notification group)

If a rule triggers an alarm, an email is sent to the members of the notification group specified when the rule was scheduled. For more information about scheduling rules see Section 2.3.2, “Scheduling Rules”.

You can set the subject line of the email that is sent by editing the **Subject Line** field. You can insert the alarm level using the string `{0}`, the rule name that was triggered using `{1}`, and the server name that triggered it using `{2}`.

You can optionally set the emails that are generated when the notification occurs to be limited to the size of an SMS message. You can then use a notification group email address that forwards to an SMS service so that you can get notification on your phone with an SMS service. Selecting the checkbox next to **SMS** only changes the length and format of the message that is sent to be compatible with the

SMS size limits.

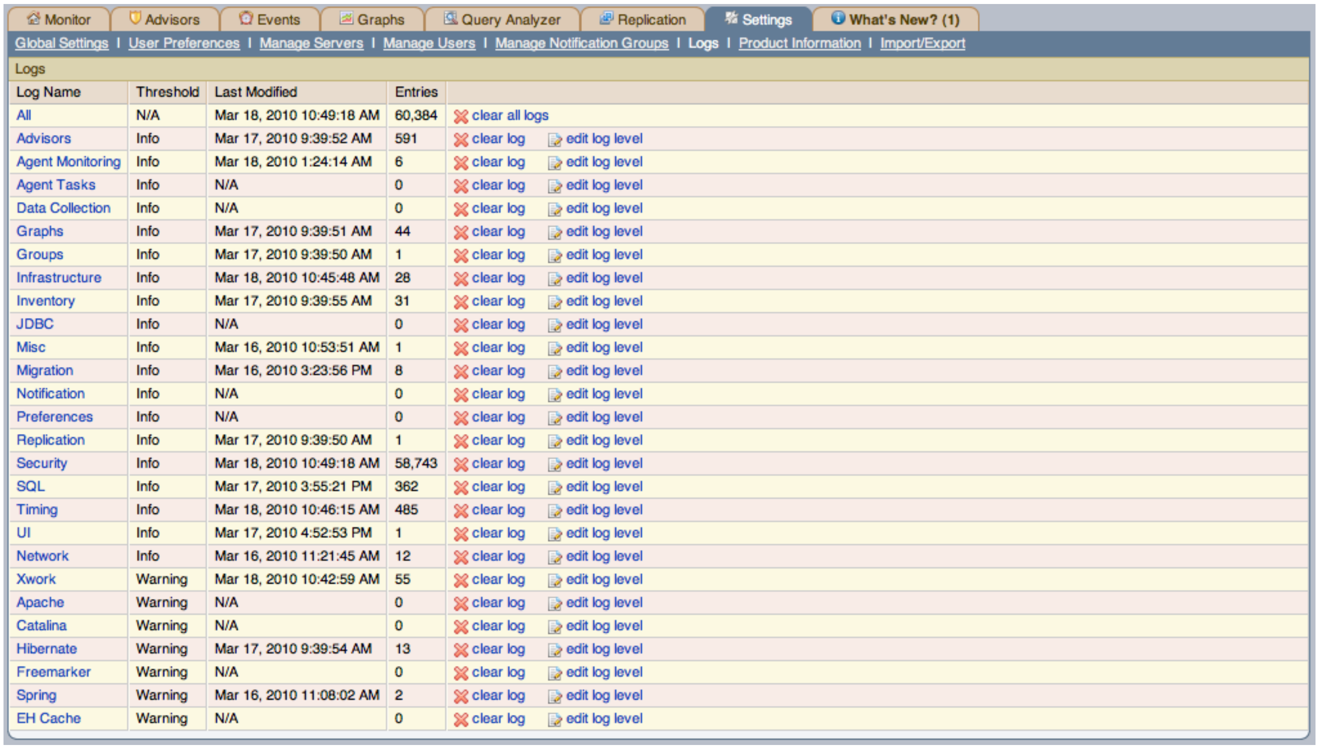
You can also optionally select a notification group to be used when sending critical error messages (such as MySQL Enterprise Agent or MySQL Enterprise Service Manager failures) by selecting the checkbox next to **MEM ADMIN**.

> **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 configure an email address with a third-party service that forwards the notification emails to an SMS number. SMS sending functionality is not provided in MySQL Enterprise Service Manager.

2.8.6. 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 2.27. MySQL Enterprise Dashboard: Logs


Monitor
Advisors
Events
Graphs
Query Analyzer
Replication
Settings
What's New? (1)
Global Settings | User Preferences | Manage Servers | Manage Users | Manage Notification Groups | Logs | Product Information | Import/Export

Logs

Log Name	Threshold	Last Modified	Entries	
All	N/A	Mar 18, 2010 10:49:18 AM	60,384	✗ clear all logs
Advisors	Info	Mar 17, 2010 9:39:52 AM	591	✗ clear log edit log level
Agent Monitoring	Info	Mar 18, 2010 1:24:14 AM	6	✗ 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	Mar 17, 2010 9:39:51 AM	44	✗ clear log edit log level
Groups	Info	Mar 17, 2010 9:39:50 AM	1	✗ clear log edit log level
Infrastructure	Info	Mar 18, 2010 10:45:48 AM	28	✗ clear log edit log level
Inventory	Info	Mar 17, 2010 9:39:55 AM	31	✗ clear log edit log level
JDBC	Info	N/A	0	✗ clear log edit log level
Misc	Info	Mar 16, 2010 10:53:51 AM	1	✗ clear log edit log level
Migration	Info	Mar 16, 2010 3:23:56 PM	8	✗ 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	Mar 17, 2010 9:39:50 AM	1	✗ clear log edit log level
Security	Info	Mar 18, 2010 10:49:18 AM	58,743	✗ clear log edit log level
SQL	Info	Mar 17, 2010 3:55:21 PM	362	✗ clear log edit log level
Timing	Info	Mar 18, 2010 10:46:15 AM	485	✗ clear log edit log level
UI	Info	Mar 17, 2010 4:52:53 PM	1	✗ clear log edit log level
Network	Info	Mar 16, 2010 11:21:45 AM	12	✗ clear log edit log level
Xwork	Warning	Mar 18, 2010 10:42:59 AM	55	✗ clear log edit log level
Apache	Warning	N/A	0	✗ clear log edit log level
Catalina	Warning	N/A	0	✗ clear log edit log level
Hibernate	Warning	Mar 17, 2010 9:39:54 AM	13	✗ clear log edit log level
Freemarker	Warning	N/A	0	✗ clear log edit log level
Spring	Warning	Mar 16, 2010 11:08:02 AM	2	✗ 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 opens 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. A confirmation dialog box lets you 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 [29].

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 [29]).

Selecting `Error` from the list box creates 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 2.8.7, “Product Information”.

2.8.7. Product Information

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 relates to your subscription settings and the servers currently being monitored. For more information, see Section 2.8.7.1, “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 number 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 provided in this section is listed below:

  - **PENDING HEARTBEATS**
  - **PROCESSED HEARTBEATS**
  - **PENDING JOBS**
  - **NUMBER OF AGENTS**
  - **MONITORED MYSQLD INSTANCES**
  - **BUILD VERSION**

  - **Advisor Version**

  - **Number of Rules**

  - **Number of Graphs**

  - **Last Successful Email**

  - **Last Email Failure**

  - **Last Failure Message**

  You can monitor the status of the email messages sent by MySQL Enterprise Service Manager by looking at the values of the **Last Successful Email** and **Last Email Failure** sections. Any failure by MySQL Enterprise Service Manager to send an email is reported, with the failure date recorded in **Last Email Failure**. You can examine the **Last Failure Message** field to determine the precise reason for the failure.

- **Readme**

  The **Readme** section contains a copy of the text `Readme` file provided with each release, and provides release specific information about your MySQL Enterprise Service Manager installation.

- **Diagnostic Report**

  The **Diagnostic Report** section includes a hyperlink, **Download diagnostic report**. The Zip file contains a variety of information and log files which can be used by support to determine and diagnose some problems with your MySQL Enterprise Monitor installation. The file is named according to your license number and the current date/time, for example `support-1234-6578-20100331T1026.zip`. The file includes, but is not limited to, the following items:

  - `Replication 1.dot`: The calculated MySQL server replication structure.

  - `catalina.out`: The Tomcat log file.

  - `java-threads.dot`: A list of the current Java threads and the dependencies.

  - `java.props`: The current Java configuration properties.

  - `java.threads`: A list of the current Java threads and their backtrace.

  - `mysql-monitor.log`: The MySQL log file.

  - `preferences.properties`: The MySQL Enterprise Monitor preferences.

  - `query.instanceOverview.html`: An HTML list of the current query analyzer data.

  - `root.csv`: A copy of your main MySQL Enterprise Monitor log file.

  - `server.props`: A copy of your server properties.

  - `subscription.xml`: Details of your subscription information.

  This report is especially useful for debugging the MySQL Enterprise Service Manager and the MySQL Enterprise Agent.

  When you select the report, a new page opens showing the report generation status. A message is provided once the generation has completed.

2.8.7.1. 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 2.3.1, “Installing and Updating Advisors”.

2.8.8. Import/Export

The **IMPORT/EXPORT** system lets you export or delete custom rules and graphs that have been imported into the system so that you can transfer them to another MySQL Enterprise Dashboard instance. You can also use this screen to import new custom rules and graphs.

To export rules or graphs:

1. Select the rules or graphs that you want to export. You can select multiple items by selecting the checkbox against each rule or graph.

2. Click EXPORT.

The custom rules and graphs are saved using a filename of the form `custom-export-DATE-TIME.jar`, where `DATE` is the date of export in international format (`YYYYMMDD`) and `TIME` is the time (`HHMM`).

To delete rules or graphs:

1. Select the rules or graphs that you want to export. You can select multiple items by selecting the checkbox against each rule or graph.

2. Click DELETE.

3. A confirmation dialog asks you to confirm that the rule or graph should be deleted. Data collections and the corresponding rules are unscheduled from all servers on which the rules have been applied. You cannot undo the operation; to re-enable the rules and graphs at a later date, export them first, then delete them so that you can re-import the custom graphs.

4. To confirm deletion, click DELETE. To cancel deletion, click CANCEL.

You can import custom rules and graphs by using the import function. You can use an XML file, or a previously exported `jar` file.

To import custom rules or graphs:

1. Click CHOOSE FILE. This presents you with a custom dialog so that you can choose a file on your client computer to be uploaded to the MySQL Enterprise Dashboard. Locate the file, and click CHOOSE.

2. Click IMPORT.

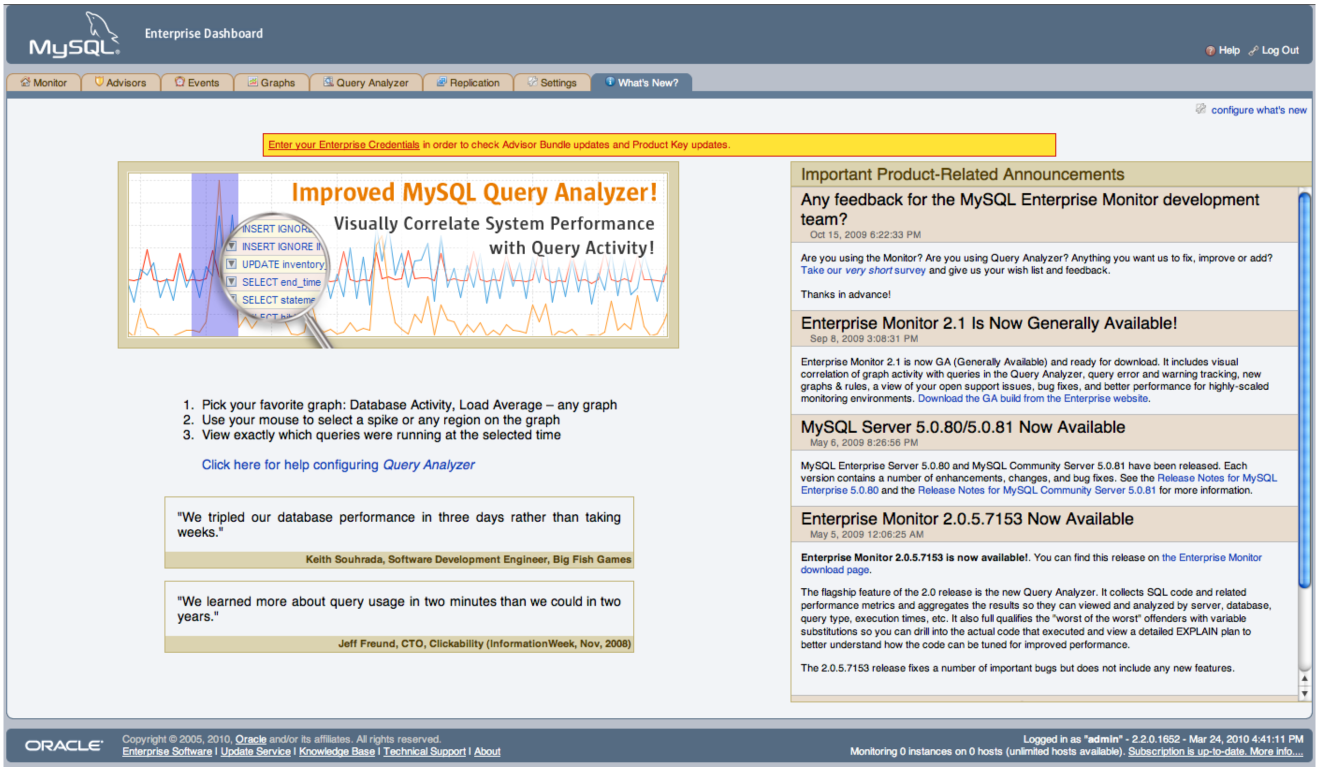
3. A notification window appears indicating the number of graphs and rules imported.

For more information on writing new rules, custom data items, and creating new graphs, see Chapter 4, *Customizing MySQL Enterprise Monitor*.

2.9. The WHAT'S NEW Tab

The What's New Page provides a simplified interface for providing updates and news related to your MySQL Enterprise Subscription. By default, the information is automatically updated every hour. To disable the updates, or force an update, see Section 2.9.1, "Setting What's New Configuration".

Figure 2.28. The What's New Tab



Enter your Enterprise Credentials in order to check Advisor Bundle updates and Product Key updates.

1. Pick your favorite graph: Database Activity, Load Average – any graph
2. Use your mouse to select a spike or any region on the graph
3. View exactly which queries were running at the selected time

Click here for help configuring *Query Analyzer*

"We tripled our database performance in three days rather than taking weeks."

Keith Souhrada, Software Development Engineer, Big Fish Games

"We learned more about query usage in two minutes than we could in two years."

Jeff Freund, CTO, Clickability (InformationWeek, Nov, 2008)

Important Product-Related Announcements

Any feedback for the MySQL Enterprise Monitor development team?
Oct 15, 2009 6:22:33 PM

Are you using the Monitor? Are you using Query Analyzer? Anything you want us to fix, improve or add? Take our *very short* survey and give us your wish list and feedback.

Thanks in advance!

Enterprise Monitor 2.1 Is Now Generally Available!
Sep 8, 2009 3:08:31 PM

Enterprise Monitor 2.1 is now GA (Generally Available) and ready for download. It includes visual correlation of graph activity with queries in the Query Analyzer, query error and warning tracking, new graphs & rules, a view of your open support issues, bug fixes, and better performance for highly-scaled monitoring environments. Download the GA build from the Enterprise website.

MySQL Server 5.0.80/5.0.81 Now Available
May 6, 2009 8:26:56 PM

MySQL Enterprise Server 5.0.80 and MySQL Community Server 5.0.81 have been released. Each version contains a number of enhancements, changes, and bug fixes. See the Release Notes for MySQL Enterprise 5.0.80 and the Release Notes for MySQL Community Server 5.0.81 for more information.

Enterprise Monitor 2.0.5.7153 Now Available
May 5, 2009 12:06:25 AM

Enterprise Monitor 2.0.5.7153 is now available!. You can find this release on the Enterprise Monitor download page.

The flagship feature of the 2.0 release is the new Query Analyzer. It collects SQL code and related performance metrics and aggregates the results so they can viewed and analyzed by server, database, query type, execution times, etc. It also full qualifies the "worst of the worst" offenders with variable substitutions so you can drill into the actual code that executed and view a detailed EXPLAIN plan to better understand how the code can be tuned for improved performance.

The 2.0.5.7153 release fixes a number of important bugs but does not include any new features.

Copyright © 2005, 2010, Oracle and/or its affiliates. All rights reserved.
Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as "**admin**" - 2.2.0.1652 - Mar 24, 2010 4:41:11 PM
Monitoring 0 instances on 0 hosts (unlimited hosts available). Subscription is up-to-date. More info....

The **WHAT'S NEW** tab is divided into three main sections:

- The area on the left-hand side of the page is used to provide information about the major new features and functionality in MySQL Enterprise Monitor.

- **OPEN SUPPORT ISSUES** provides a quick interface to the list of currently open issues with MySQL Support as identified through the MySQL Enterprise and Support websites. Issues that are currently marked in the system as *Waiting On Customer* are also added to the main MySQL Enterprise Dashboard above the Heat Chart.

  To use the Open Support Issues functionality, you must have set your MySQL Enterprise credentials within MySQL Enterprise Dashboard. For more information, see Section 2.8.1, "Global Settings".

- **IMPORTANT PRODUCT RELEASE ANNOUNCEMENTS** provides a list of news items from the news feed provided on the MySQL Enterprise website. The news feed will contain important information, including, but not limited to:

  - New releases of MySQL Products, including MySQL Server and MySQL Enterprise Monitor.
  - End-of-life and support changes.
  - Urgent errors and notifications.

  When items are new (unread), they will be indicated with a star icon. Unread items on the What's New page will also be notified with a number in the tab title within the main Dashboard. Items are automatically marked as read when you visit the What's New page.

The announcements information is made available in two ways, using either a connection to the MySQL Enterprise website, when the information will be updated at a set interval, or through information included with the MySQL Enterprise Advisor bundles.

Note

To use poll the MySQL Enterprise site directly your MySQL Enterprise Service Manager must have access to the MySQL Enterprise website. If your MySQL Enterprise Service Manager is not directly connected to the Internet, you can set an HTTP Proxy to be used to access the information. For more information on configuring an HTTP Proxy, see Section 2.8.1, "Global Settings".

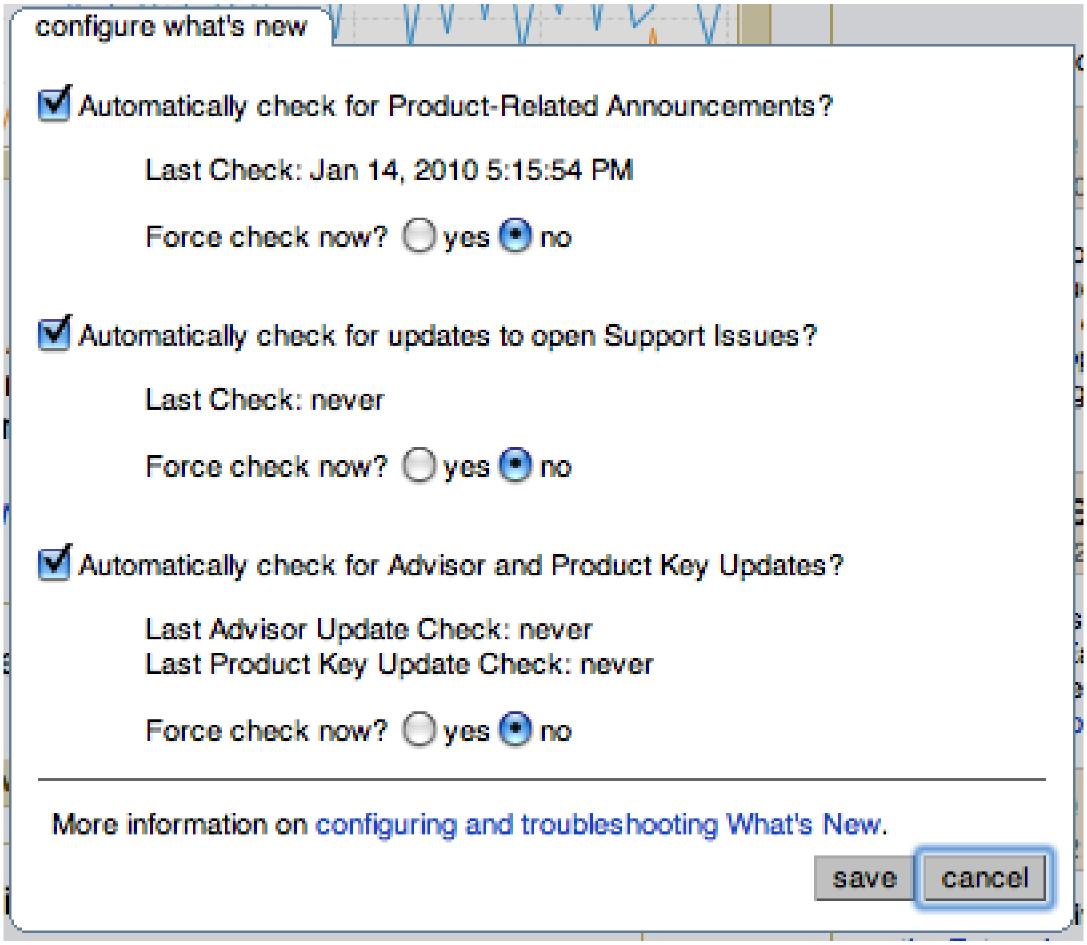
Note

You can disable the display of your open Support issues and product notifications by using the CONFIGURE WHAT'S NEW link on the **WHAT'S NEW** tab. For more information, see Section 2.9.1, "Setting What's New Configuration".

2.9.1. Setting What's New Configuration

You can configure the operation and forcibly update the information on the What's New page by clicking the CONFIGURE WHAT'S NEW button at the top-right of the What's New page.

Figure 2.29. What's New Configuration



configure what's new

☑ Automatically check for Product-Related Announcements?

Last Check: Jan 14, 2010 5:15:54 PM

Force check now? ○ yes ● no

☑ Automatically check for updates to open Support Issues?

Last Check: never

Force check now? ○ yes ● no

☑ Automatically check for Advisor and Product Key Updates?

Last Advisor Update Check: never
Last Product Key Update Check: never

Force check now? ○ yes ● no

More information on configuring and troubleshooting What's New.

The configurable options are:

- **AUTOMATICALLY CHECK FOR NEW ANNOUNCEMENTS?**

  To disable the New Announcements updates, deselect the checkbox. Updates are enabled by default.

  The last time an update of the New Announcements information was successfully executed is displayed below the title.

  You can force an update of the New Announcements information by selecting the YES radio button next to **FORCE CHECK NOW?**. A check will be executed when you click the SAVE button to close the configuration dialog.

- **AUTOMATICALLY CHECK TO SUPPORT ISSUES?**

  To disable the Support Issues updates, deselect the checkbox. Updates are enabled by default.

  The last time an update of the Support Issues information was successfully executed is displayed below the title.

  You can force an update of the Support Issues information by selecting the YES radio button next to **FORCE CHECK NOW?**. A check will be executed when you click the SAVE button to close the configuration dialog.

- Automatically check for Advisor and Product Key Updates?

  The MySQL Enterprise Service Manager can automatically look for changes to your MySQL Enterprise Advisor and product key information by polling the MySQL Enterprise website.

  To disable the Advisor and Product key updates, deselect the checkbox. Updates are enabled by default.

  You can force a check for updates to the Advisor and Product key information by selecting the YES radio button next to **FORCE CHECK NOW?**. A check will be executed when you click the SAVE button to close the configuration dialog.

Click SAVE to save your settings. If you selected any system to force an update, the update will take place in the background. To cancel your changes, click CANCEL.

Chapter 3. Using the Query Analyzer

Query Analyzer lets you monitor SQL statements executed on a MySQL server and see details about each query, as well as number of executions and execution times. Similar queries with different literal values are combined ("normalized") for reporting purposes.

Query Analyzer collects information about SQL statements that a MySQL client application sends to the MySQL server. The client application must either route its database requests through the MySQL Enterprise Agent proxy service, or use the MySQL Enterprise Monitor Plugin for a given connector that sends the information directly to MySQL Enterprise Service Manager. Once your MySQL client application is configured to communicate via the MySQL Enterprise Agent, queries are monitored and the normalized queries are sent to the MySQL Enterprise Agent.

For the different ways to enable Query Analysis, see Section 3.1, "Providing Query Analyzer Data".

> **Important**
>
> The MySQL Proxy component and Query Analyzer require that clients connecting through MySQL Enterprise Agent use MySQL 5.0 or later. Clients that use the library provided with MySQL 4.1 or earlier do not work with MySQL Enterprise Agent.

Once the data is collected, you view and monitor the queries, check the execution statistics, and filter and drill down on the information. By comparing the queries to the server graphs, you can correlate query execution with server status. For more information on viewing, filtering and reporting on the query analyzer data, see Section 3.2, "Query Analyzer User Interface".

> **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.

3.1. Providing Query Analyzer Data

The Query Analyzer can be fed information from a number of different sources, in each case with the provider supplying the statistical information about the queries, execution times, result counts and other data to be displayed and analyzed with the Query Analyzer tab within MySQL Enterprise Dashboard.

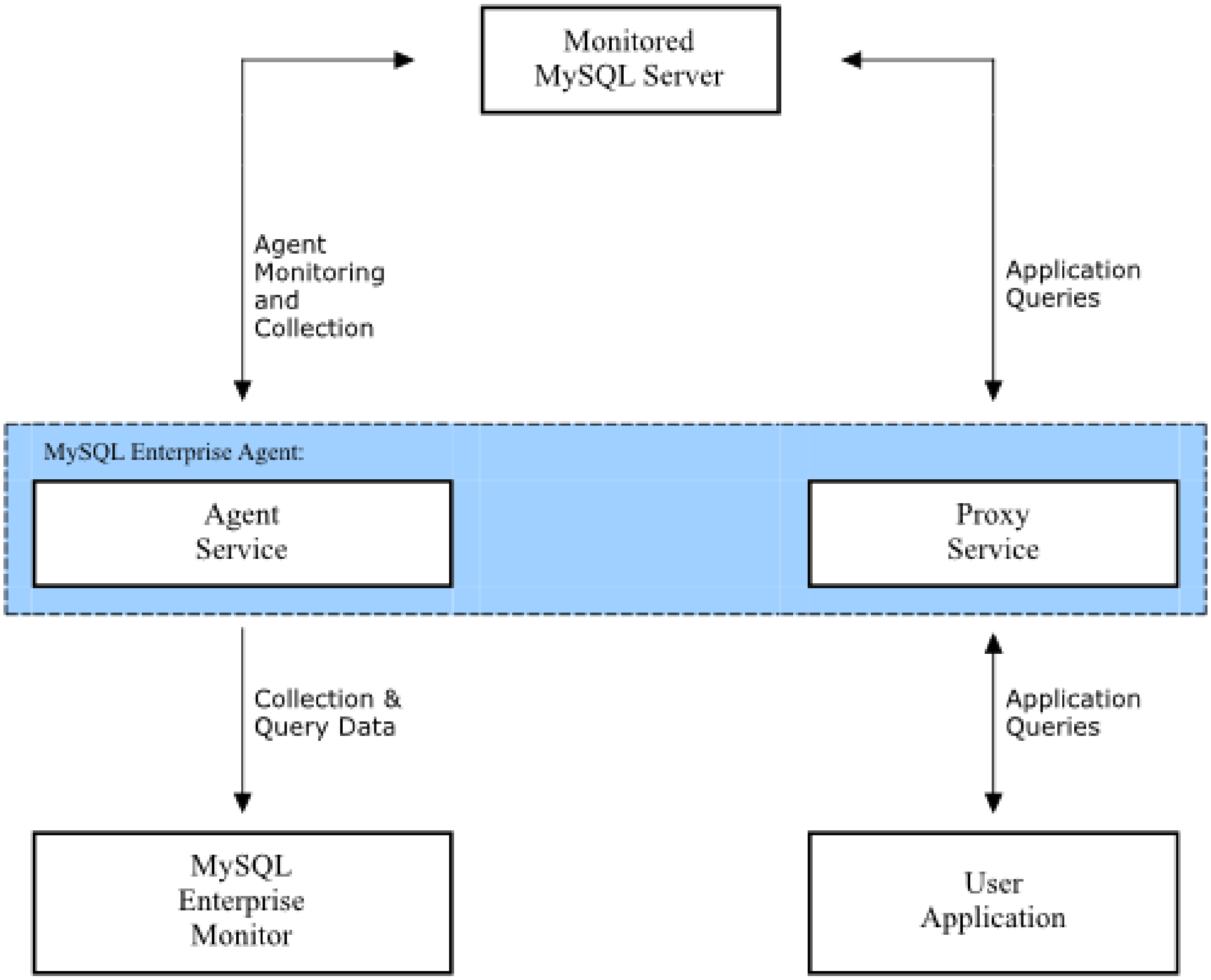
There are a number of different methods available for supplying query information to MySQL Enterprise Service Manager:

- Using the MySQL Proxy functionality built into MySQL Enterprise Agent. This is the method offered and supported since MySQL Enterprise Monitor 2.0.

  When using the proxy, queries from the client application are directed through the proxy to the MySQL server, with all results from the MySQL server being returned from the proxy to the client application. While this transmission of information takes place, the proxy component keeps statistical information and supplies this information to MySQL Enterprise Service Manager for analysis.

  The structure of this process in relation to the MySQL server containing the client application data, the client application, and the the MySQL Enterprise Monitor components is shown in the figure below.

  Figure 3.1. MySQL Enterprise Monitor Architecture using Query Analyzer with Proxy



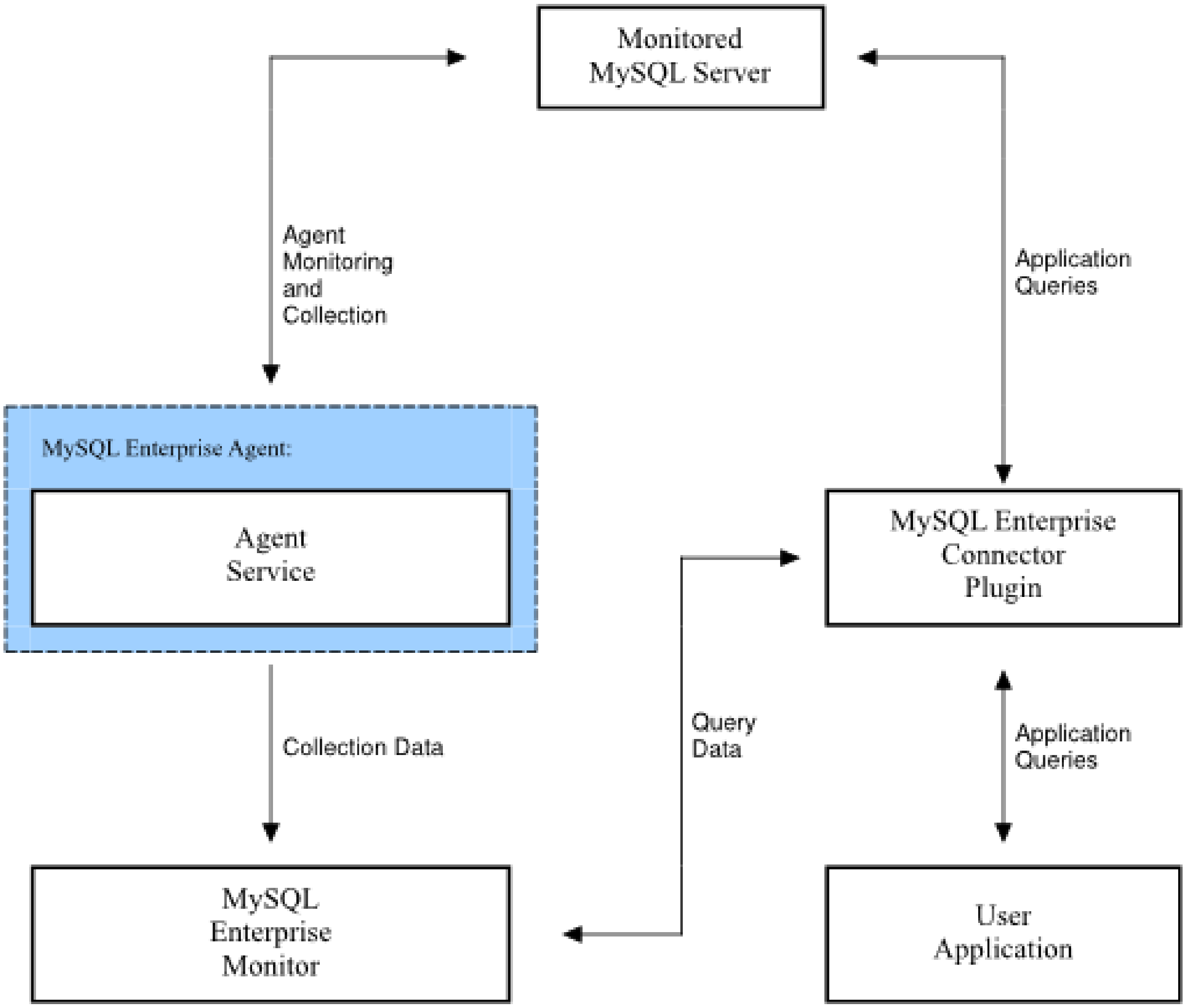
For more information on using this method, see Section 3.1.3, “Using the Agent Proxy”.

- Use a MySQL connector with a corresponding MySQL Enterprise Monitor Plugin that provides tracing and statistical information directly to MySQL Enterprise Service Manager.

  Using this method requires a connector that is capable of collecting and sending the query statistical data directly to MySQL Enterprise Service Manager. The connectors handles all of the collection of the basic query statistics, such as the execution time for each query, and the row counts, and provides this information to MySQL Enterprise Service Manager for analysis. This solution negates the need the proxy component, providing you with maximum speed between your application, connector and MySQL server containing your client application data.

  The structure of this solution is shown in the figure below.

Figure 3.2. MySQL Enterprise Monitor Architecture using Query Analyzer Directly from Connector



For more information on using this method, see the table below for information on which connectors support this method of operation.

The table below provides information on which connectors support the different functionality types.

Table 3.1. Query Analyzer Feeding Solution Comparison

Feature	C/Java	C/NET	Proxy
Uses Proxy	No	No	Yes
Supports EXPLAIN	Yes	Yes	Yes

3.1.1. Using the MySQL Enterprise Plugin for Connector/J

The MySQL Enterprise Plugin for Connector/J enables any application using the Connector/J JDBC driver to automatically supply query analyzer information directly to MySQL Enterprise Service Manager. Once you have installed the installed plugin library in combination with your existing Connector/J driver, information will be sent to the MySQL Enterprise Service Manager for analysis. You do not need to modify or alter your applications to enable this functionality.

Table 3.2. Query Analyzer Connector/J Summary

Feature	C/Java
Uses Proxy	No
Supports EXPLAIN	Yes

Prerequisites

1. MySQL Connector/J version 5.1.12 or later
2. JDK-1.5.0 or later
3. MySQL Enterprise Service Manager version 2.2 or later
4. The MySQL instance that is being used must by monitored by a MySQL Enterprise monitoring agent. The `mysql.inventory` table must be accessible to the configured user for the Connector/J connection to get the UUID of the monitored server.
5. Apache Commons logging in the `CLASSPATH` of the application being analyzed. If Commons Logging is not in use, this product ships with a version that must be deployed in the Application's `CLASSPATH`.

Installation

Place the jar file `lib/c-java-mysql-enterprise-plugin-[version].jar` in the application's `CLASSPATH` where it is visible to the version of MySQL Connector/J in use. This location is preferably in the same location as MySQL Connector/J's jar file, or in a parent classloader to MySQL Connector/J's jar file's location.

If the application being analyzed does not have Apache Commons Logging in the `CLASSPATH`, install the file `lib/required/commons-logging-1.1.1.jar` in the application's `CLASSPATH` as well. If no other component in your application uses Apache Commons Logging, install it in the same place where the query analyzer plugin was installed.

Configuration

Once the required jar files are deployed, the connection URL for jdbc needs to be modified to use and configure the query analyzer plugin. The following properties and their corresponding values need to be added to the URL that is used by the application to be analyzed.

For more information on Connector/J properties, see Connector/J (JDBC) Reference.

Table 3.3. MySQL Monitor Plugin for Connector/J Options

Property Name	Property Type	Default
`mysqlenterprise.httpSocketTimeoutMillis`	numeric	
`mysqlenterprise.serviceManagerPassword`	string	
`mysqlenterprise.serviceManagerUrl`	string	
`mysqlenterprise.serviceManagerUser`	string	
`statementInterceptors`	string	

- `statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters`

  Enables the operation of the plugin. Without this configuration option, query information will not be provided to MySQL Enterprise Service Manager

- `mysqlenterprise.serviceManagerUrl`

Version Introduced	2.2.0	
	Permitted Values	
	Type	`string`

Specifies the URL to the service manager. You can obtain this information from the configuration file created when MySQL Enterprise Service Manager was installed. Note that you must include the full URL, including the port number. For example `http://monitor.example.com:18080`.

- `mysqlenterprise.serviceManagerUser`

Version Introduced	2.2.0	
	Permitted Values	
	Type	`string`

Specifies the agent username to be used when communicating information to MySQL Enterprise Service Manager.

- `mysqlenterprise.serviceManagerPassword`

Version Introduced	2.2.0	
	Permitted Values	
	Type	`string`

Specifies the agent password to be used when communicating information to MySQL Enterprise Service Manager.

- **Optional**
- `mysqlenterprise.httpSocketTimeoutMillis`

Version Introduced	2.2.0	
	Permitted Values	
	Type	`numeric`

The number of milliseconds to wait when communicating with the service manager.

If you want to use SSL to communicate query information to the MySQL Enterprise Service Manager then you can configure this using the properties below. Out of the box, the plugin is configured to work with the self-signed certificates that ship with the enterprise monitor. Therefore, it will provide confidentiality for the information that is sent by the plugin if the service manager URL starts with `https://`. However, as-configured, the plugin does not prevent man-in-the-middle attacks nor does it support client-side certificates for authentication. To enable these features, refer to the non-default values of the following parameters:

Table 3.4. MySQL Monitor Plugin for Connector/J SSL Options

Property Name	Property Type	Default
`mysqlenterprise.clientCertificateKeystorePassword`	string	
`mysqlenterprise.clientCertificateKeystoreType`	string	
`mysqlenterprise.clientCertificateKeystoreUrl`	string	
`mysqlenterprise.trustCertificateKeystorePassword`	string	
`mysqlenterprise.trustCertificateKeystoreType`	numeric	

Property Name	Property Type	Default
mysqlenterprise.trustCertificateKeystoreUrl	string	
mysqlenterprise.verifySslCerts	boolean	false
mysqlenterprise.verifySslHostnames	boolean	false

- mysqlenterprise.verifySslHostnames

  If set to "true", host names presented in the SSL server certificate will be verified following the scheme listed at BrowserCompatHostnameVerifier which is the same scheme as used by Firefox and Curl and specified by RFC 2818.

- mysqlenterprise.verifySslCerts

  Specifies whether the plugin verify that the certificate presented by the server was signed by a CA in the trustCertificateKeystore.

- mysqlenterprise.trustCertificateKeystoreUrl

  URL to the trusted root certificate KeyStore (if not specified, use Java defaults).

- mysqlenterprise.trustCertificateKeystorePassword

  Password for the certificate store.

- mysqlenterprise.trustCertificateKeystoreType

  =[type] KeyStore type for trusted root certificates (NULL or empty means use JKS default, standard keystore types supported by the JVM are JKS and PKCS12, your environment may have more available depending on what security products are installed and available to the JVM.

- mysqlenterprise.clientCertificateKeystoreUrl

  URL to the client certificate KeyStore (if not specified, use Java defaults)

- mysqlenterprise.clientCertificateKeystorePassword=[password]

  Password for the client certificate store.

- mysqlenterprise.clientCertificateKeystoreType

  KeyStore type for client certificates (NULL or empty means use JKS default, standard keystore types supported by the JVM are JKS and PKCS12, your environment may have more available depending on what security products are installed and available to the JVM.

For example, to configure a Connector/J to communicate with the MySQL Enterprise Service Manager localhost, on port 18080, using the agent username agent, and password PASSWORD, you would add the following options to your connection string:

```
statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters»
    &serviceManagerUrl=http://localhost:18080/»
    &serviceManagerUser=agent»
    &serviceManagerPassword=PASSWORD
```

Remember that you must also add the application specific options to the JDBC URL. For example, the following fragment will connect to the MySQL database test on localhost, using the user and password of mysqltest, while also collecting query data and sending it to the MySQL Enterprise Service Manager on localhost:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

Connection conn = null;
...
try {
    conn =
       DriverManager.getConnection("jdbc:mysql://localhost/test?" +
                                   "user=mysqltest" +
                                   "&password=mysqltest" +
                                    "&statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters" +
                                    "&serviceManagerUrl=http://localhost:18080/" +
                                    "&serviceManagerUser=agent" +
                                    "&serviceManagerPassword=PASSWORD"
                                    );

    // Do something with the Connection

   ...
} catch (SQLException ex) {
    // handle any errors
}
```

> **Note**
>
> If a DataSource is in use (typically when using Glassfish, Weblogic, or Websphere), these properties must be passed as part of the URL property, they can not be added to the DataSource configuration itself.

If an alternate logging system has not been selected for Connector/J, it is recommended that Connector/J's log factory is configured to use something other than the standard logger by adding the following property to the URL or DataSource:

- logger=Log4JLogger (for applications using Log4J)
- logger=CommonsLogger (for applications using Apache Commons Logging)
- logger=JDK14Logger (for applications using Java 1.4 or later logging)

Further configuration of the plugin is done via the Enterprise Service Manager's user interface, using the Query Analyzer tab. From here, the capture of query performance data for a given MySQL instance that the plugin is being with can be enabled or disabled. Further, the level of detail that is captured; summary, examples (with source code locations) and `EXPLAIN` plans can be configured as well.

The query analyzer plugin checks the configuration once each minute (while it is reporting data it has collected), thus configuration changes are not immediate.

Usage Notes

You should be aware of the following while using the MySQL Enterprise Plugin for Connector/J:

- When a Java application that is reporting queries terminates, you may lose the information about some queries because the queued query information has not been sent to MySQL Enterprise Service Manager. You can get round this issue by adding a sleep to your application to ensure the normal reporting schedule submits the outstanding queries. For example, you could add:

  ```
  Thread.sleep(60000);
  ```

  To your application before the application terminates.

3.1.2. Using the MySQL Enterprise Plugin for Connector/NET

The MySQL Enterprise Plugin for Connector/NET provides query analyzer data from any application using Connector/NET, including both standalone and web based applications. Information about the queries is sent directly to the MySQL Enterprise Service Manager. Once you have installed the MySQL Enterprise Plugin for Connector/NET, query analysis will be available for your applications, without requiring any modification to your application code.

Table 3.5. Query Analyzer Connector/NET Summary

Feature	C/NET
Uses Proxy	No
Supports EXPLAIN	Yes

Prerequisites

- You must be using Connector/NET version 6.2.3 or later.
- Your application should already be using the `Mysql.data.dll` and have been built with the library requirement.
- If you are using the released builds of Connector/NET, you must include the `logging=true` option within your connection string.

Installation

Download the MySQL Enterprise Plugin for Connector/NET package. Extract the package using a suitable Zip tool, and then place the plugin library, `MySql.MonitorPlugin.dll` in the same directory as your compiled application.

Configuration

If the application that you want to profile does not have an `app.config` application configuration file then you need to make one.

To enable query analyzer functionality, you need to register the trace listeners in the `System.Diagnostics` section of the `app.config` file. You can see an example configuration file below:

```
<system.diagnostics>

  <sources>

    <source name="mysql" switchName="SourceSwitch" switchType="System.Diagnostics.SourceSwitch">

      <listeners>

        <add name="EMTrace" type="MySql.EMTrace.EMTraceListener, MySql.MonitorPlugin"
                initializeData=""
                Host="SERVERHOST:SERVERPORT"
                PostInterval="POSTINTERVAL"
                UserId="AGENTUSERID"
                Password="AGENTPASSWORD"/>

      </listeners>

    </source>

  </sources>

  <switches>


    <add name="SourceSwitch" value="All"/>

  </switches>

</system.diagnostics>
<system.data>

  <DbProviderFactories>

    <add name="MySQL Data Provider" invariant="MySql.Data.MySqlClient" description=".Net Framework Data Provider for MySQL
.MySqlClientFactory, MySql.Data, Version=6.2.1.0, Culture=neutral, PublicKeyToken=c5687fc88969c44d" />

  </DbProviderFactories>

</system.data>
```

Within the configuration, you must set the values of the following parameters:

- `Host`

  The hostname and port number, separated by a colon, of the MySQL Enterprise Service Manager where the query analyzer data should be sent.

- `PostInterval`

  The interval in seconds between posts of the collected query analyzer information from you application to the MySQL Enterprise Service Manager. You should choose the value for this carefully. Too long and it may take some time for queries to appear in the query analyzer.

- `UserId`

  The username of a user within MySQL Enterprise Service Manager that has rights to send agent information.

- `Password`

  The password of a user within MySQL Enterprise Service Manager that has rights to send agent information.

To get extended information on queries and have that information sent up to MySQL Enterprise Monitor you must enable the Connector/NET usage advisor. Examples of the extended information includes not using an index on a query, or not accessing all columns from a result set.

To enable the usage advisor, you must add `usage advisor=true` to the connection string within your application. Enabling this option also automatically enables logging within Connector/NET. For more information, see Connector/NET Connection String Options Reference.

During execution of the application during development within Visual Studio, you should see a significant amount of output in the **Output** window. If you are running the application standalone and want to monitor the trace output, you should configure and additional listener in your configuration file. You can do this by adding the following within the `system.diagnostics` section of your `app.config` file:

```
<trace autoflush="false" indentsize="4">
  <listeners>
    <add name="consoleListener" type="System.Diagnostics.ConsoleTraceListener" />
  </listeners>
</trace>
```

3.1.3. Using the Agent Proxy

There are three different ways of feeding query analyzer using the MySQL Enterprise Agent Proxy Service:

- 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 3.1.3.1, “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 3.1.3.2, “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=:6446
  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 C.2.1, “MySQL Enterprise Agent Configurable Options”.

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 hostname, 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';
  ```

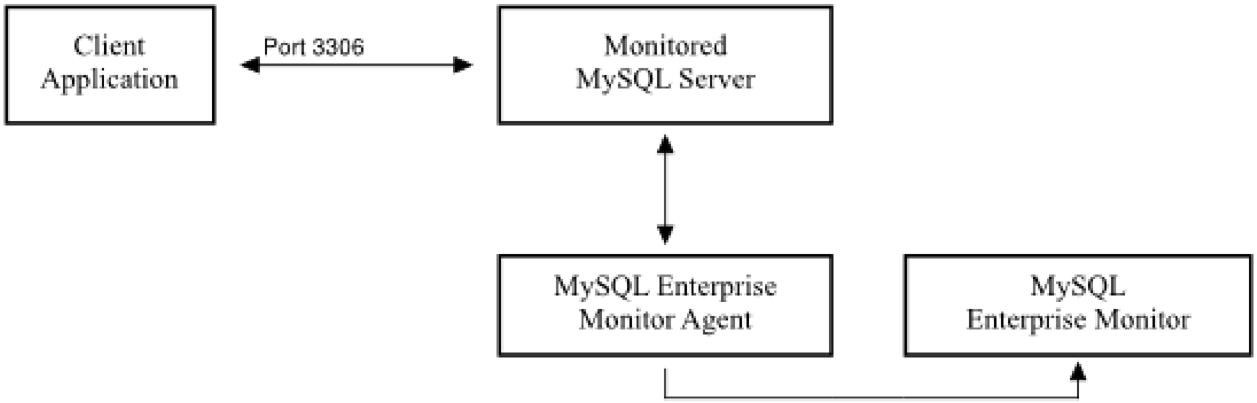
- The MySQL client application user must have `SELECT` privileges on the `mysql.inventory` table. This table contains the server UUID and is required to report the query analyzer data to the MySQL Enterprise Service Manager. To enable this, use the `GRANT` option:

  ```
  mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
  ```

3.1.3.1. Enabling Query Analyzer by Changing the MySQL Client Application

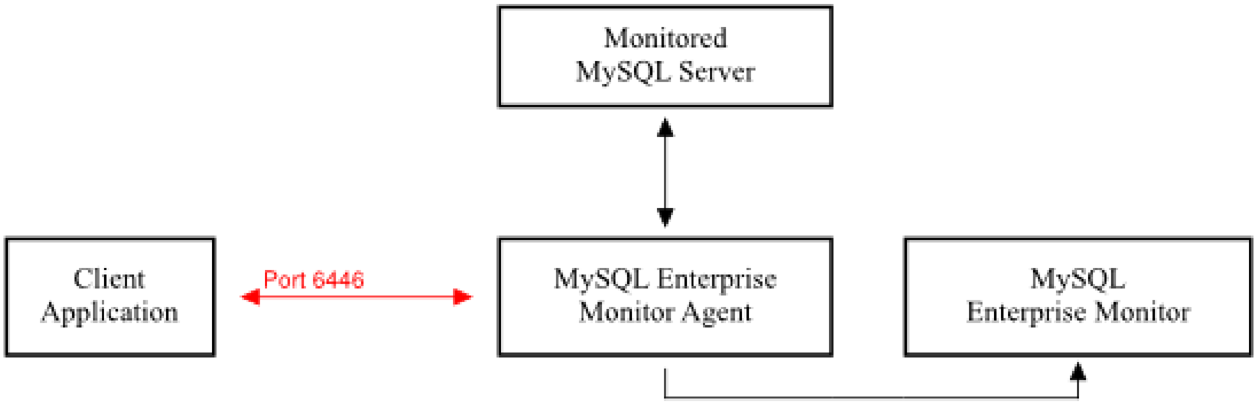
Generally, changing your MySQL client application is the easiest and recommended method. For example, given a typical structure like the one shown in the figure below, the client application would need to be modified so that it no longer communicated directly with the MySQL server, but to the agent/proxy.

Figure 3.3. MySQL Enterprise Dashboard: Standard Agent/Monitor Topology



You can see an example of the structure when communicating via the agent/proxy below.

Figure 3.4. 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 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 via 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=6446 --protocol=tcp
> shell> mysql --port=6446 --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.

3.1.3.2. 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 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 Agent.

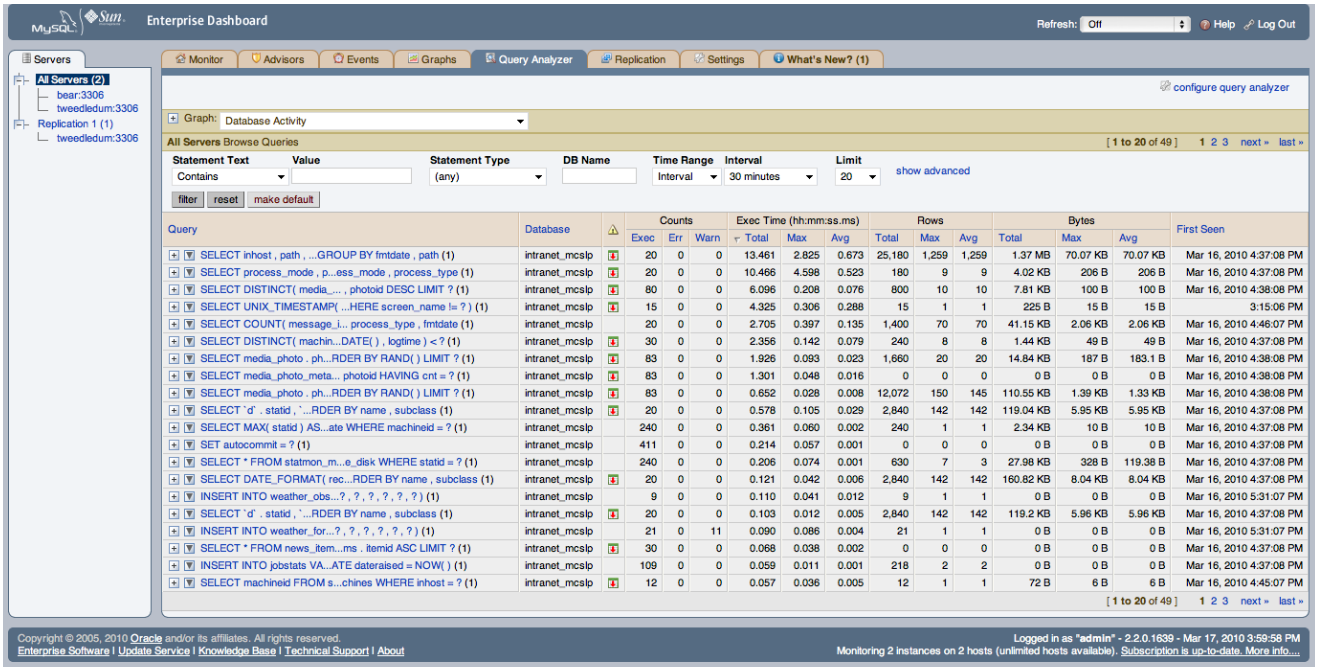
You should now be able to connect to your MySQL server through the MySQL Enterprise Agent by connecting on the original port:

```
shell> mysql --host=127.0.0.1
```

3.2. Query Analyzer User Interface

To analyse the queries captured by the agent/proxy, change to the **Query Analyzer** tab. You can see an example of the table on that page in the figure below.

Figure 3.5. MySQL Enterprise Dashboard: Query Analyzer



Graph: Database Activity

All Servers Browse Queries

Statement Text: Contains; Value: ; Statement Type: (any); DB Name: ; Time Range: Interval; Interval: 30 minutes; Limit: 20

Query	Database	⚠	Counts			Exec Time (hh:mm:ss.ms)			Rows			Bytes			First Seen
			Exec	Err	Warn	Total	Max	Avg	Total	Max	Avg	Total	Max	Avg	
SELECT inhost , path , ...GROUP BY fmtdate , path (1)	intranet_mcslp	⬇	20	0	0	13.461	2.825	0.673	25,180	1,259	1,259	1.37 MB	70.07 KB	70.07 KB	Mar 16, 2010 4:37:08 PM
SELECT process_mode , p...ess_mode , process_type (1)	intranet_mcslp	⬇	20	0	0	10.466	4.598	0.523	180	9	9	4.02 KB	206 B	206 B	Mar 16, 2010 4:37:08 PM
SELECT DISTINCT(media_... , photoid DESC LIMIT ? (1)	intranet_mcslp	⬇	80	0	0	6.096	0.208	0.076	800	10	10	7.81 KB	100 B	100 B	Mar 16, 2010 4:38:08 PM
SELECT UNIX_TIMESTAMP(...HERE screen_name != ?) (1)	intranet_mcslp	⬇	15	0	0	4.325	0.306	0.288	15	1	1	225 B	15 B	15 B	3:15:06 PM
SELECT COUNT(message_i... process_type , fmtdate (1)	intranet_mcslp		20	0	0	2.705	0.397	0.135	1,400	70	70	41.15 KB	2.06 KB	2.06 KB	Mar 16, 2010 4:46:07 PM
SELECT DISTINCT(machin...DATE( ) , logtime) < ? (1)	intranet_mcslp	⬇	30	0	0	2.356	0.142	0.079	240	8	8	1.44 KB	49 B	49 B	Mar 16, 2010 4:37:08 PM
SELECT media_photo . ph...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	⬇	83	0	0	1.926	0.093	0.023	1,660	20	20	14.84 KB	187 B	183.1 B	Mar 16, 2010 4:38:08 PM
SELECT media_photo_meta... photoid HAVING cnt = ? (1)	intranet_mcslp	⬇	83	0	0	1.301	0.048	0.016	0	0	0	0 B	0 B	0 B	Mar 16, 2010 4:38:08 PM
SELECT media_photo . ph...RDER BY RAND( ) LIMIT ? (1)	intranet_mcslp	⬇	83	0	0	0.652	0.028	0.008	12,072	150	145	110.55 KB	1.39 KB	1.33 KB	Mar 16, 2010 4:38:08 PM
SELECT `d` . statid , `...RDER BY name , subclass (1)	intranet_mcslp	⬇	20	0	0	0.578	0.105	0.029	2,840	142	142	119.04 KB	5.95 KB	5.95 KB	Mar 16, 2010 4:37:08 PM
SELECT MAX(statid) AS...ate WHERE machineid = ? (1)	intranet_mcslp		240	0	0	0.361	0.060	0.002	240	1	1	2.34 KB	10 B	10 B	Mar 16, 2010 4:37:08 PM
SET autocommit = ? (1)	intranet_mcslp		411	0	0	0.214	0.057	0.001	0	0	0	0 B	0 B	0 B	Mar 16, 2010 4:37:08 PM
SELECT * FROM statmon_m...e_disk WHERE statid = ? (1)	intranet_mcslp		240	0	0	0.206	0.074	0.001	630	7	3	27.98 KB	328 B	119.38 B	Mar 16, 2010 4:37:08 PM
SELECT DATE_FORMAT(rec...RDER BY name , subclass (1)	intranet_mcslp	⬇	20	0	0	0.121	0.042	0.006	2,840	142	142	160.82 KB	8.04 KB	8.04 KB	Mar 16, 2010 4:37:08 PM
INSERT INTO weather_obs...? , ? , ? , ? , ? , ?) (1)	intranet_mcslp		9	0	0	0.110	0.041	0.012	9	1	1	0 B	0 B	0 B	Mar 16, 2010 5:31:07 PM
SELECT `d` . statid , `...RDER BY name , subclass (1)	intranet_mcslp	⬇	20	0	0	0.103	0.012	0.005	2,840	142	142	119.2 KB	5.96 KB	5.96 KB	Mar 16, 2010 4:37:08 PM
INSERT INTO weather_for...? , ? , ? , ? , ? , ?) (1)	intranet_mcslp		21	0	11	0.090	0.086	0.004	21	1	1	0 B	0 B	0 B	Mar 16, 2010 5:31:07 PM
SELECT * FROM news_item...ms . itemid ASC LIMIT ? (1)	intranet_mcslp	⬇	30	0	0	0.068	0.038	0.002	0	0	0	0 B	0 B	0 B	Mar 16, 2010 4:37:08 PM
INSERT INTO jobstats VA...ATE dateraised = NOW( ) (1)	intranet_mcslp		109	0	0	0.059	0.011	0.001	218	2	2	0 B	0 B	0 B	Mar 16, 2010 4:37:08 PM
SELECT machineid FROM s...chines WHERE inhost = ? (1)	intranet_mcslp	⬇	12	0	0	0.057	0.036	0.005	12	1	1	72 B	6 B	6 B	Mar 16, 2010 4:45:07 PM

Copyright © 2005, 2010 Oracle and/or its affiliates. All rights reserved.
Enterprise Software I Update Service I Knowledge Base I Technical Support I About

Logged in as "admin" - 2.2.0.1639 - Mar 17, 2010 3:59:58 PM
Monitoring 2 instances on 2 hosts (unlimited hosts available). Subscription is up-to-date. More info....

The main Query Analyzer table provides the summary information for all of the queries executed via the agent/proxy. The table will track all the queries submitted to the server via 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 3.2.4, "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.

- **EXECUTION NOTICES**: Highlights any specific issues experienced when running queries, including excessive table scans and bad index usage. These provide an immediate indication of a problem with a query that may require additional examination.

- **COUNTS**: The number of times that the query has been executed. The column is sub-divided into three further columns, showing the number of executions of the query (**EXEC**), the number of times an error has been reported for the query (**ERR**), and the number of times a warning has been produced (**WARN**).

- **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 **EXEC TIME** column is further subdivided into the following columns:

  - **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 Using Query Analyzer Data.

  > **Note**
  >
  > Due to limitations in the counters used for monitor queries on Microsoft 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 date and time the query was first 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.

To help you and locate queries you can filter the list of queries using a variety of criteria. For more information on the filtering methods available, see Section 3.2.4, “Filtering Query Analyzer Data”.

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, hover over the triangle CREATE ALIAS link against the query. The maximum length for a query alias is 255 characters.

To get more detailed information about reported queries and their execution, see Using Query Analyzer Data.

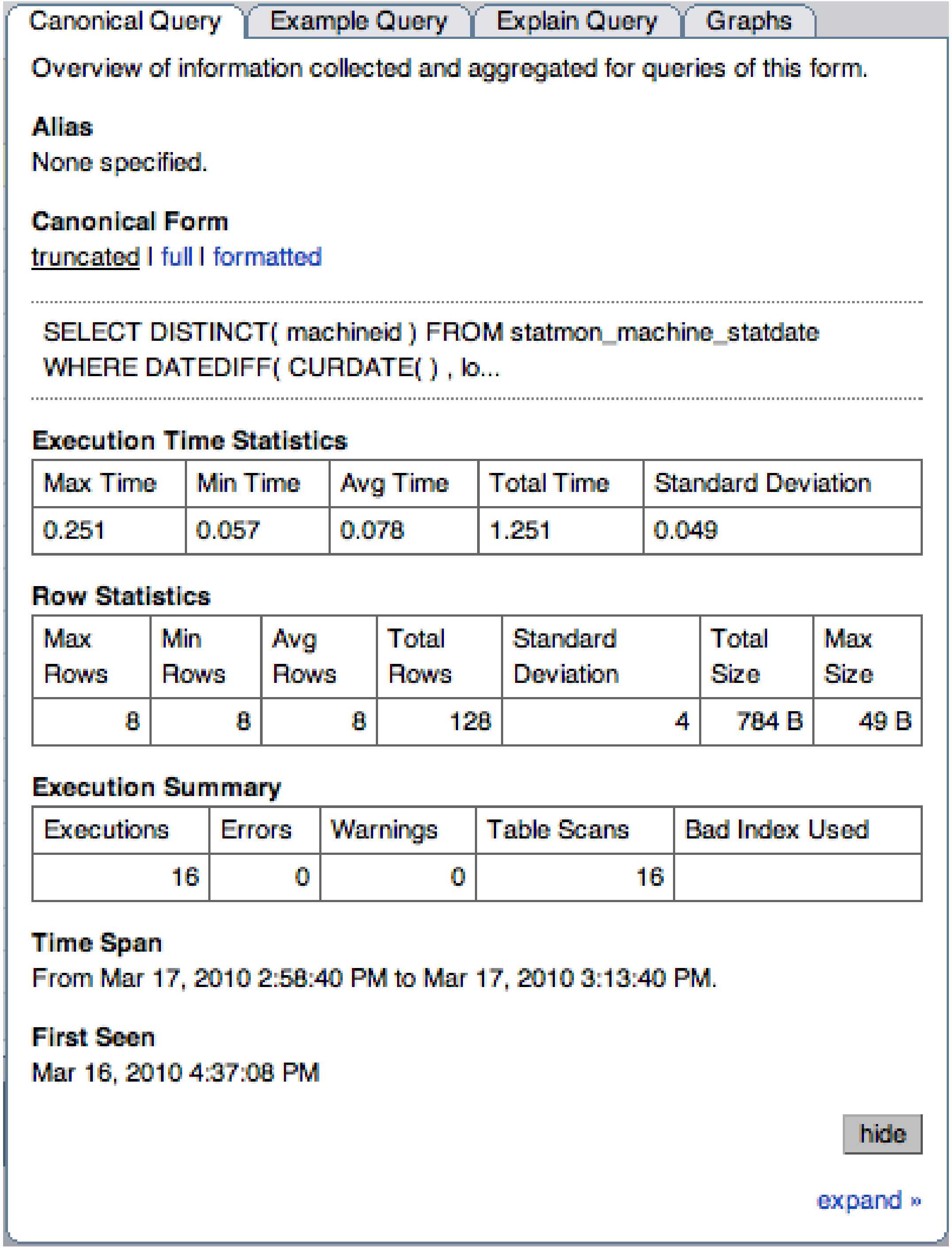
3.2.1. Getting Detailed Query Information

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 **CANONICAL QUERY** view.

If enabled, you may also view **EXAMPLE QUERY**, which provide more detailed data about a specific query, including the data and parameters submitted. In addition, you may also enable **EXPLAIN QUERY**, which provides you with the ability to remotely execute an `EXPLAIN` statement with the specified query and view the resulting information. Finally, you can also view any graph data produced during the execution of the query by using the **GRAPHS** tab.

- The **Canonical Query** tab:

  Figure 3.6. 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

```
SELECT DISTINCT( machineid ) FROM statmon_machine_statdate
WHERE DATEDIFF( CURDATE( ) , lo...
```

Execution Time Statistics

Max Time	Min Time	Avg Time	Total Time	Standard Deviation
0.251	0.057	0.078	1.251	0.049

Row Statistics

Max Rows	Min Rows	Avg Rows	Total Rows	Standard Deviation	Total Size	Max Size
8	8	8	128	4	784 B	49 B

Execution Summary

Executions	Errors	Warnings	Table Scans	Bad Index Used
16	0	0	16	

Time Span

From Mar 17, 2010 2:58:40 PM to Mar 17, 2010 3:13:40 PM.

First Seen

Mar 16, 2010 4:37:08 PM

The canonical view for a query provides three different views of the query, which can be changed using the links under the **Canonical Form** section. 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.

In addition to the summary information given in the table, the **Execution Time Statistics** section provides you with more detailed execution time 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.

The **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 **Time Span** header.

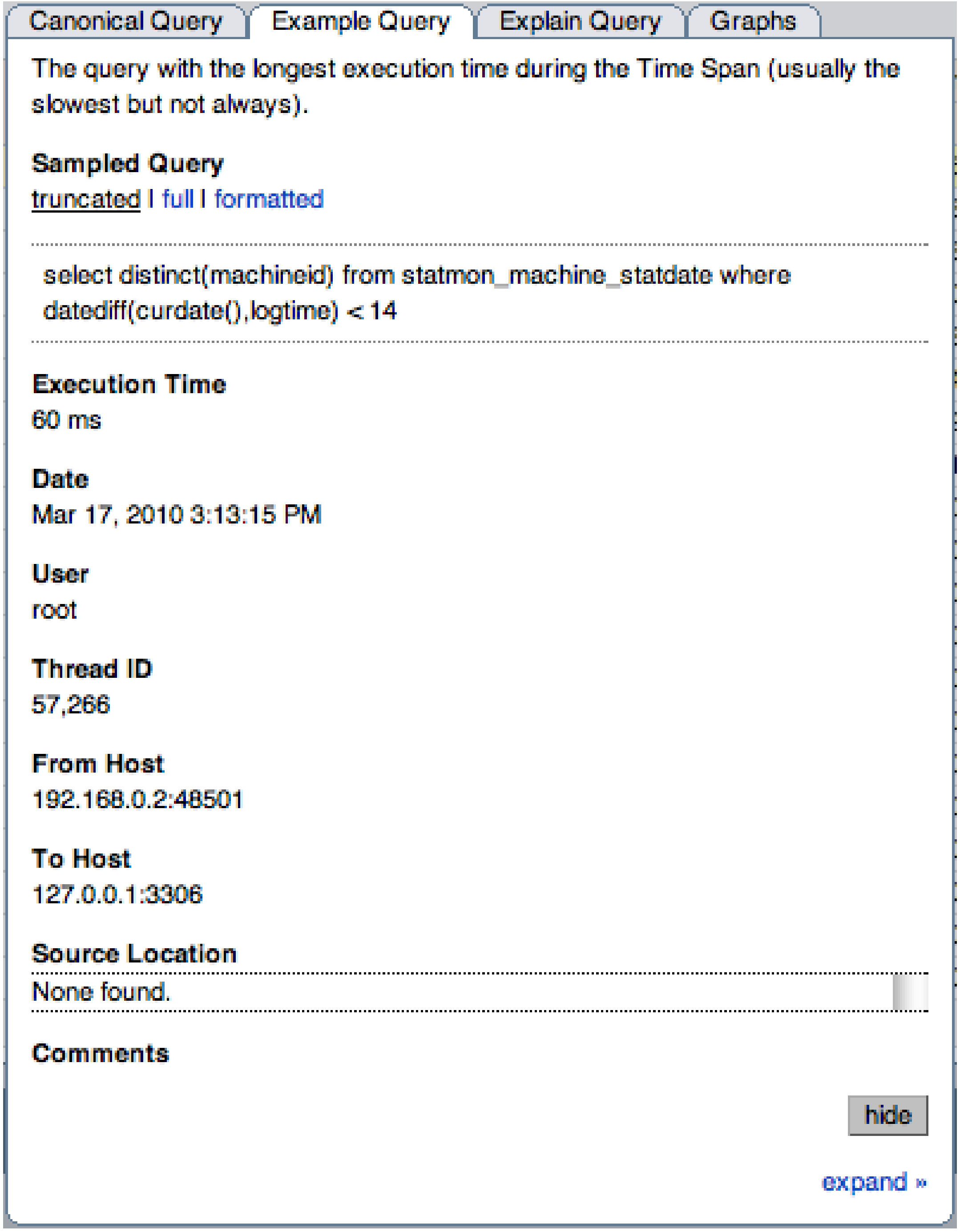
The **Execution Summary** section provides the summary data available in the main table, covering the execution count, and counts of the number of errors, warnings, queries that triggered table scans, and those that indicated a bad index use.

The **First Seen** reports when the query was first seen, whether or not this was within the indicated **Time Span**.

To close the query detail window, click the Hide button.

- The **Example Query** tab:

Figure 3.7. 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

```
select distinct(machineid) from statmon_machine_statdate where
datediff(curdate(),logtime) < 14
```

Execution Time

60 ms

Date

Mar 17, 2010 3:13:15 PM

User

root

Thread ID

57,266

From Host

192.168.0.2:48501

To Host

127.0.0.1:3306

Source Location

None found.

Comments

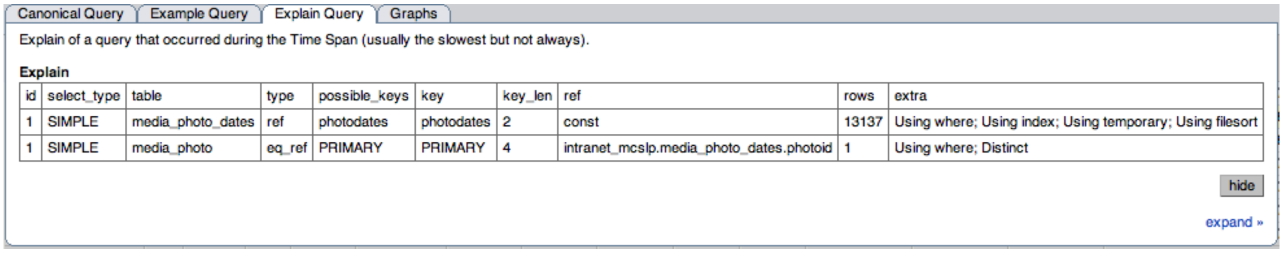
The **Example Query** tab provides detailed information about the most expensive query executed, as determined by the execution time.

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.

For queries from any of the MySQL Enterprise Plugin for Connectors, the **Source Location** will contain the information from the Connector of where the query was generated.

- The **Explain Query** tab:

Figure 3.8. 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).

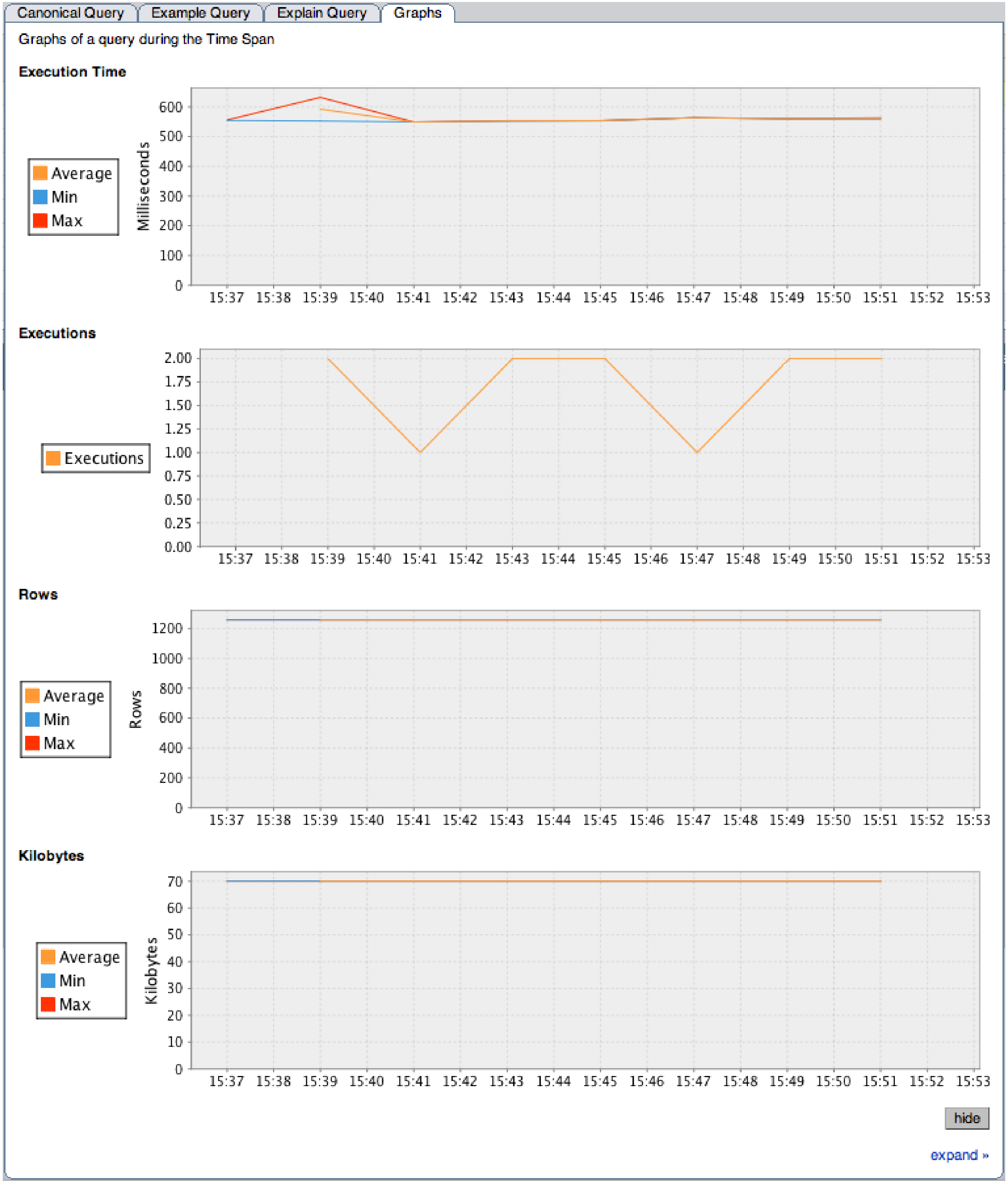
Explain

id	select_type	table	type	possible_keys	key	key_len	ref	rows	extra
1	SIMPLE	media_photo_dates	ref	photodates	photodates	2	const	13137	Using where; Using index; Using temporary; Using filesort
1	SIMPLE	media_photo	eq_ref	PRIMARY	PRIMARY	4	intranet_mcslp.media_photo_dates.photoid	1	Using where; Distinct

The **Explain Query** tab lets you view the output from running the query with the `EXPLAIN` prefix. For more information, see `EXPLAIN` Syntax.

- The **Graphs** tab:

Figure 3.9. MySQL Enterprise Dashboard: Graphs Tab for a Query



The **Graphs** tab shows key graphs over the selected time period for the example query. Shown are graphs of the **Execution Time**, **Executions**, **Rows**, and **Kilobytes**. These can be used to identify deviations from the normal values. Unlike the query-correlation graphs, these graphs shown only the query statistics over the given time period.

3.2.2. Using Graphs to Identify Queries

The MySQL Enterprise Dashboard supports correlated graphs so that you can compare the graphed execution parameters, such as the server load, thread statistics, or RAM usage, against the queries that were being executed by the server or servers being monitored at that time.

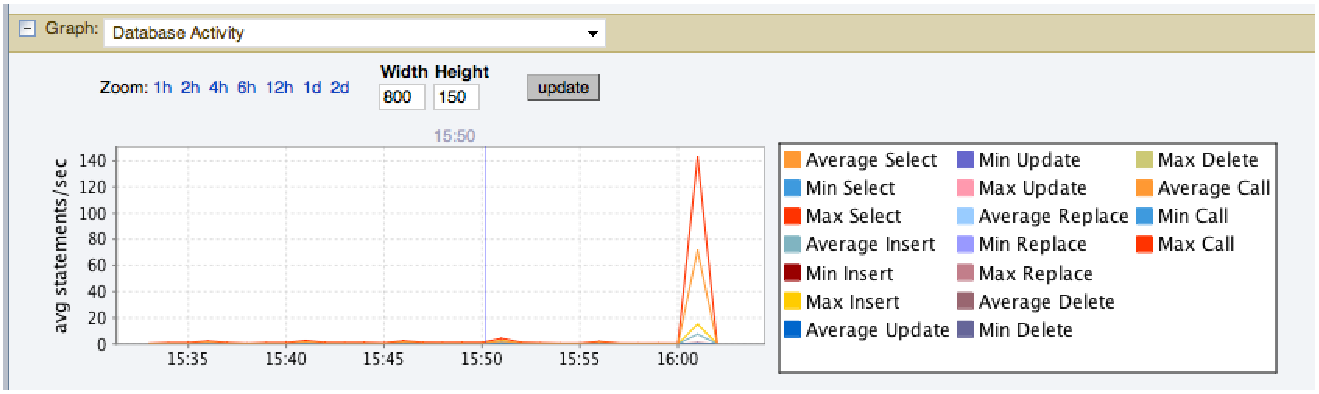
You can use the correlated graphs in two different ways:

- By dragging and selecting an area on a graph as displayed within the **Monitor** over view tab, or the **Graphs** tab of the MySQL Enterprise Dashboard. When using either of these two tabs, you can drag and select any region on a displayed graph, and the display will change to the **Query Analyzer** tab, displaying the corresponding zoomed graph, and the associated queries being executed during the selected period.
- You can select a graph to be displayed within the **Query Analyzer** tab. Open the **Graph** section, and use the pop-up list to select the graph to be displayed. You can see an example of this in Figure 3.10, "MySQL Enterprise Dashboard: Correlated Graphs".

When using the correlated graphs, selecting an area within the graph will set the start and end time within the query filtering selection. You can combine with other filtering options, such as the **Query Type**, to zero-in on the queries that you want to examine.

To use the graphs in this manner, select a starting point and click, while holding down the button, drag a selection area to set the time-range for the query display. The time range that you have selected will be displayed above the graph as you select the area.

Figure 3.10. MySQL Enterprise Dashboard: Correlated Graphs



Graph: Database Activity

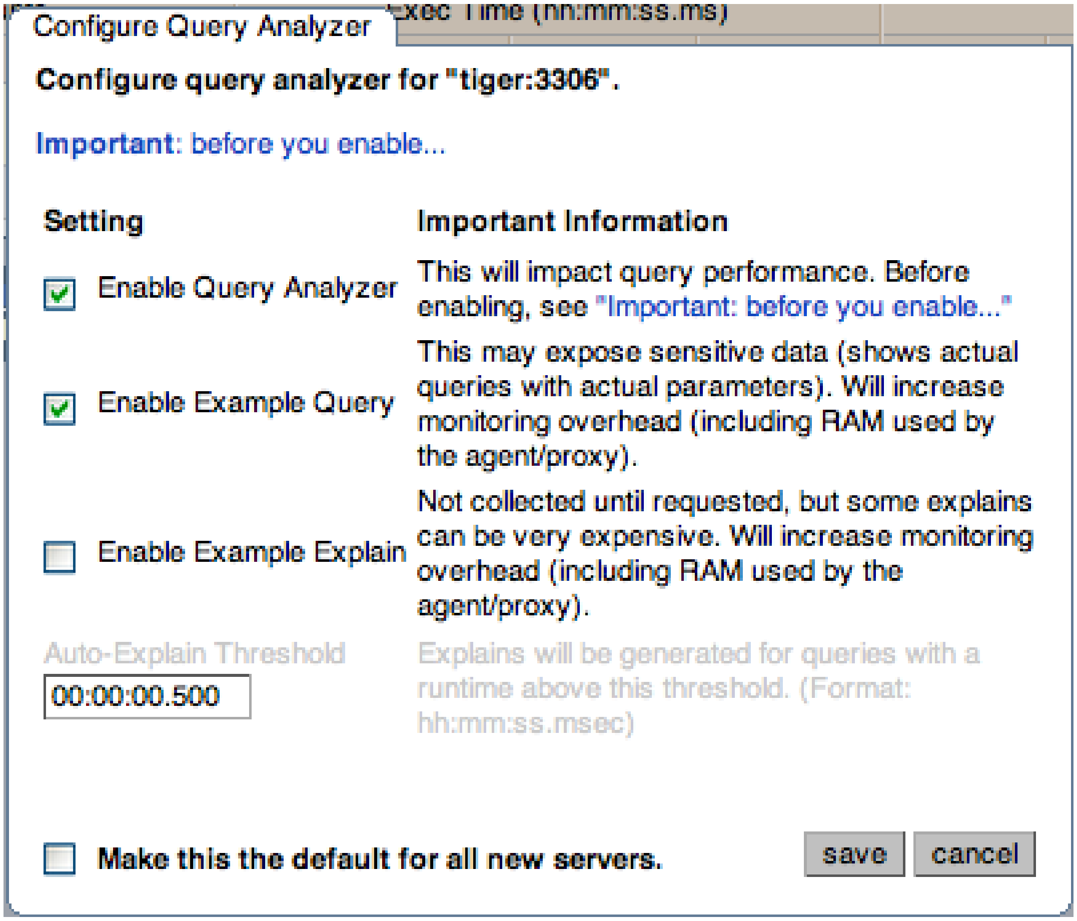
Width 800 Height 150

You can also explicitly zoom in an out of the displayed graph by using the links next to the **Zoom** label. You can select to show just an hour of the graph material, or larger ranges up to two days. At all times, the area on the graph highlighted in blue is the actual date/time range selected within the list of analyzed queries.

3.2.3. 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 3.11. MySQL Enterprise Dashboard: Query Analyzer Configuration



Configure Query Analyzer

Configure query analyzer for "tiger: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).
Auto-Explain Threshold 00:00:00.500	Explains will be generated for queries with a runtime above this threshold. (Format: hh:mm:ss.msec)

□ **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, select the check box.

  Note

  The Query Analyzer functionality may show as being enabled on a server, even though the modules within MySQL Enterprise Agent supplying query analyzer data may not have been enabled.

  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 3.2.1, "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 F, *MySQL Enterprise Monitor Frequently Asked Questions*.

To enable or disable query analyzer for an individual server, go to the **Settings** tab 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** tab. 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** tab.

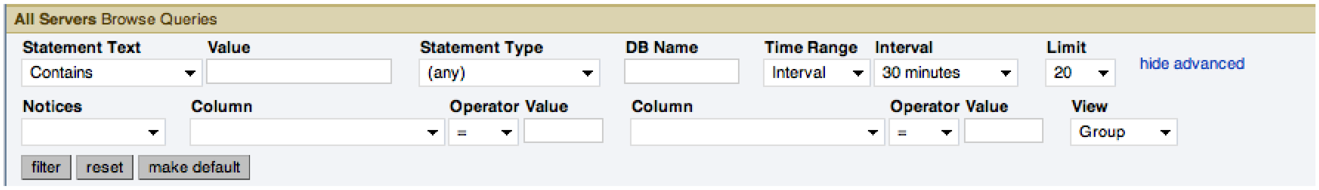
3.2.4. Filtering Query Analyzer Data

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:

The filtering functionality is available in a simple format, supporting simple statement and timing based filtering, and an advanced option allowing you to filter by specific columns within the query analyzer table.

Figure 3.12. MySQL Enterprise Dashboard: Query Analyzer Filter Options



All Servers Browse Queries

Statement Text: Contains
Value:
Statement Type: (any)
DB Name:
Time Range: Interval
Interval: 30 minutes
Limit: 20
Notices:
Column:
Operator Value: =
Column:
Operator Value: =
View: Group

The basic filter options are:

- **Statement Text** and **Value** support text searching of the normalized query. For the search type you can specify either a basic text match (**Contains**), or a regular expression match (**Regex**). 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.

- **Statement Type**: Limits the search to statements of a particular type (`SELECT`, `LITERAL`, etc.).
- **DB Name**: Limits 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 informa-

tion, see Pattern Matching.

- The **Time Range** 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** time range lets you 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.

- **Limit** specifies the number of queries to be displayed within each page.

To use the advanced filtering techniques, click SHOW ADVANCED. This provides additional filters:

- **Notices**: Filters on the notices column, allowing you to filter the list to show only the queries that did not raise a notice, indicated a full table scan, or indicated that a bad index was used.

- Two column filters are provided, which allow you to filter the queries based on specific values within any of the columns shown in the Query Analyzer report list.

  To use the column filters, you must specify the **Column** that you want to filter on, the **Operator** to use when performing the comparison and the **Value** that you want to compare.

  For example, to filter by showing all the queries that return more than 100 rows on average, set the **Column** to `Average Rows`, the **Operator** to `>=`, and the **Value** to 100.

- 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.

All the filter settings that you specify are used collectively, that is, all the specified filter options are used to match against the list of queries.

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.

If you want to make the filter options that you have set the default when viewing the **Query Analyzer** page, click the MAKE DEFAULT button. The settings will be retained for the current user only.

3.3. 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 configuration file, `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=:6446
    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 6446 (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=:6446
  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 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 3.2.3, “Query Analyzer Settings”.

Chapter 4. Customizing MySQL Enterprise Monitor

You can customize your MySQL Enterprise Monitor rules, advisors, and graphs, based on your organization's business rules, best practices, and the types of issues you can anticipate.

4.1. Creating Advisors and Rules

For common scenarios, reuse or edit the advisors and rules provided by MySQL Enterprise. To create new advisors and rules for your own needs, go to the **Advisors** tab and choose the `Create/Edit Rule` link.

4.1.1. Creating Advisors

Similar existing rules are grouped together in advisor groups.

The built-in advisor groups are:

- Administration
- Heat Chart
- Performance
- Replication
- Schema
- Security

You can also create your own advisor group to suit your circumstances. Click the CREATE ADVISOR button. Enter an appropriate name and click the ADD button. The newly created group appears in the `Advisor` column.

The newly created advisor is added to the list box of advisors shown in Figure 2.5, "MySQL Enterprise Dashboard: Editing Rules". You can now use this category of advisors when you create a new rule.

4.1.2. Overview of Rule Creation

You create rules using the same screen seen in Figure 2.5, "MySQL Enterprise Dashboard: Editing Rules". To create a new rule with all-new settings, click the CREATE RULE button. To create a rule similar to an existing one, click the COPY RULE button. You can edit any rule element during the copying process, unlike editing an existing rule.

You can change the rule name, the advisor group that a rule belongs to and you can set your own version number. In Figure 2.5, "MySQL Enterprise Dashboard: Editing Rules", you have already seen how the threshold and frequency of a rule may be altered.

> **Note**
>
> If you do not specify a version number for the new rule, the version `1.0` is automatically added.

Most importantly, you can alter a rule's expression. Expressions are the core of a MySQL Enterprise Advisor and 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 mathematical operations.

An expression has two main characteristics:

- An expression tests whether a best practice is being violated.
- The result of an expression must always be 1 or 0 (corresponding to 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, the best practice is being followed and no alarm is raised.

For example, if you decide that enabling binary logging is a best practice for a production server (as Oracle recommends), then this best practice is 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 http://dev.mysql.com/doc/refman/5.0/en/functions.html. For a complete list of the built-in variables used to create rules, see http://dev.mysql.com/doc/refman/5.0/en/mysqld-option-tables.html.

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. To define more than one variable, click the ADD ROW button. For a complete listing of the data collection items used in creating rules, see Appendix D, *Data Collection Items*.

The remaining fields determine the information that you receive in a notification email or the informational pop-up window associated with each advisor.

> **Note**
>
> When saving a new rule, choose a unique name not used by any existing rule.

4.1.3. Variables

When MySQL Enterprise Monitor evaluates an expression, it replaces variables with 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 is 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.

Variables can be used in the `Description`, `Advice`, `Action`, and `Links` attributes of a rule, as well as in expressions. This lets you 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. If `%Key_reads%` has a value of `4522`, the message becomes “The current value of Key_reads is 4522.”

4.1.4. Thresholds

Each expression has a threshold value that triggers an alert. The `THRESHOLD` keyword associates 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 straightforward. 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. You cannot just use the expression `%log_bin% == "OFF"`, because this would not test binary logging against a threshold and so would not result in an alert.

Specify precise conditions when each expression should evaluated, to avoid false alarms. 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 first part of the expression, `(%Uptime% > 10800)`, holds off evaluating this expression until the system has been running for 10800 seconds (3 hours). 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 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)`. The rule is not evaluated unless there is plenty of activity associated with the key cache.

If there are periods of time during which you don't want a rule to be evaluated, you can define a “blackout period”. For example, the expression for the “Slave Too Far Behind Master” rule is: `%Seconds_Behind_Master% > THRESHOLD`. You might run a backup process between 6 and 7 pm on a replication slave, during which it is normal for that slave to get substantially behind the master. To avoid an alert during this blackout period, add the following to the expression: `&& CURTIME() NOT BETWEEN '18:00:00' AND '19:00:00'`.

4.1.5. Using Strings

Enclose string values within double quotation marks in the `Expression` or the `Thresholds` text boxes. For example, the expression for the “Slave I/O Thread Not Running” rule is:

```
(%Slave_running% == "ON") && (%Slave_IO_Running% != THRESHOLD)
```

Similarly, the `Critical Alerts` threshold text box is set to a value of `"Yes"`.

When the expression is evaluated, either `"OFF"` or `"ON"` is 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 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.

4.1.6. Wiki Format

When editing or defining a rule, you can enter text in Wiki format in the `Problem Description`, `Advice`, `Recommended Action`, and `Links and Further Reading` text boxes. You can format and highlight text and add hyperlinks, using the notation listed in the following table.

Table 4.1. 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.

4.1.7. Creating a New Rule: An Example

This section documents the steps to create a rule. Before creating a rule, review the preceding sections of this chapter.

This example creates a rule that checks the number of rows in a table. We decide that having 50,000 rows in this table triggers a critical alert. Lesser numbers produce 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, in this case something such as "Excessive number of records in *`table_name`* table".

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 4.1.1, "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 to monitor. Note that the variable `%`*`table_name`*`_num_rows%` is not yet 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 D, *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. Optionally, use Wiki markup for these text areas. 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.

After you create the rule, schedule it against the server that contains the database table to monitor. For instructions on scheduling rules, see Section 2.3.2, “Scheduling Rules”.

4.1.8. Creating a Custom Data Collection Item

Section 4.1.7, “Creating a New Rule: An Example” shows how to create a custom rule and Appendix D, *Data Collection Items* shows the data items you can use in rule creation. In some circumstances, you might want to create a rule that uses a custom data collection item.

To create additional custom data collection items, edit the `share/mysql-proxy/items/custom.xml` custom rule file added during installation. The `custom.xml` is installed automatically, but is not modified during an upgrade installation. The rules file is listed last in the `agent-item-files` configuration parameter, so any rules and data collection items you add to this file supercede the standard data collection definitions.

After you make changes to the `custom.xml`, restart the agent to load the custom definitions.

4.1.8.1. Modifying the 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. You can have multiple items in your `custom.xml` file, just specify additional `class` entries. The `custom.xml` file in the installation includes a sample entry. You can replace that entry, or add the following definition to the file:

```
<?xml version="1.0" encoding="utf-8"?>
<classes>
  <class>
    <classname>innodb_min_free</classname>
    <namespace>mysql</namespace>
    <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>
```

After saving the file, ensure that the `agent-item-files` parameter within the configuration includes the `custom.xml` file. For example:

```
[mysql-proxy]
 ...
agent-item-files = share/mysql-proxy/items/quan.lua,share/mysql-proxy/items/items-mysql-monitor.xml,»
      share/mysql-proxy/items/agent-allocation-stats.lua,share/mysql-proxy/items/custom.xml
 ...
```

Restart the agent to put this change into effect. To do this, see:

- Windows: Section 5.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 5.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 5.3.5.2, “Starting/Stopping the Agent on Mac OS X”

Once the agent has restarted, the new data item is in the `Data Item` drop down list box on the `Rule Definition` page. Its fully qualified name is `mysql:table:innodb_min_free`.

4.2. Customizing Standard Advisors

You can override the standard rules and the data collection items that support them by adding entries to your `custom-items.xml` with different queries to collect information. To customize data collection for existing advisors, you can use the information in Section 4.1, “Creating Advisors and Rules”.

4.3. Disabling and Unscheduling Rules

You can change a rule so that it no longer applies to a specific server or group of servers, or suspend a rule for a period of time.

To disable or unschedule an advisor, choose the `Current Schedule` screen of the **ADVISORS** tab.

You can disable or unschedule one or more rules using the buttons on the upper or lower left of the screen. You can also change one rule at a time by clicking the `enabled` or `unschedule` hyperlink to the right of a rule.

To no longer run a rule against a specific server, expand the advisor group and the specific rule by clicking the + button and then 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. To institute this rule again, use the `Add to Schedule` page.

To suspend a rule temporarily, use the DISABLE button and follow the same process as 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. To re-enable a disabled rule, click the `disabled` link or the ENABLE button.

You can alter multiple rules for one or more servers by selecting the appropriate checkbox and 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.

4.4. Advisor Blackout Periods

During maintenance periods for database servers, you can stop Monitor Agents from reporting their findings. During such 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.

To enable a blackout period, enter 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
```

Check the `configuration_report.txt` file for the host name and port to use. Specify the correct port for the Tomcat server. Specify the server to blackout using the name that appears in the Server Tree, including the colon and port number as shown in the preceding example.

When the HTTP authentication dialog box requesting your Dashboard user name and password opens, 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.

To confirm that a server is blacked out, check that its name is greyed out in the Dashboard.

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 does **not** reactivate a blacked out server.

4.4.1. Scripting Blackouts

You can write a script to black out a server, rather than opening a web browser and typing entries into the address bar. 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 initial `#!` 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`**.

Check the `configuration_report.txt` file for the host name and port to use. Specify the correct port for the Tomcat server. Specify the server to black out using the name that appears in the Server Tree, including the colon and port number as shown in the preceding example. Specify the name of a user who is a "manager". Specifying a user with "dba" rights only will not black out a server and no error will be displayed.

To confirm that a server is blacked out, check that its name is greyed out in the Dashboard. To end the blackout, run the same script, changing the final argument to `false`.

Note

Restarting MySQL Enterprise Monitor does **not** reactivate a blacked out server.

Chapter 5. MySQL Enterprise Monitor Installation Reference

> **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 Agents. 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 can 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 MySQL operator 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 5.2, “Service Manager Installation”.

2. Configure the Service Manager, see Section 5.3, “Monitor Agent Installation”.

3. Start the MySQL Enterprise Service Manager and MySQL Enterprise Agent instances, and then use the MySQL Enterprise Dashboard to install the Advisors and complete the configuration and installation. See Section 5.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

Depending on your configuration and environment, you download various 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 have names like `mysqlmonitor-2.1.0.1096-linux-x86_64-installer.bin`, with the appropriate version and platform name. To perform an upgrade, download the upgrade installer, with a name like `mysqlmonitor-2.1.0.1096-linux-x86_64-update-installer.bin`.

- One or more MySQL Enterprise Agent, one for each MySQL Server that you want to monitor. Download an installer package for the right platform for the MySQL server you want to manage. Agent installers are available with names like `mysqlmonitoragent-2.1.0.1093-linux-debian3.1-powerpc-installer.bin`. Upgrade installers to update an existing MySQL Enterprise Agent installation are named like `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 5.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

- *Optional*

  An advisor bundle equal to the level of your MySQL Enterprise subscription (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 5.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

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

5.1. 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.

5.1.1. 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 in order 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 5.3.1, "Creating a MySQL User Account for the Monitor Agent".

5.1.2. 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`

5.1.3. 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 5.1.1, "Existing Users".

5.2. 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, to determine the version numbers of the components, enter `http://server_name:18080/main?command=list_versions` into the web browser's address bar.

During installation, versions of MySQL and Tomcat are 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 can choose different values. The installer detects ports that are already in use and lets you 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 confirm during installation. Before you start your installation, review the section on these common parameters, then proceed to section specific to your installation platform. For details of the common parameters, see Section 5.2.1, “Service Manager Installation Common Parameters”. For information on installation under Windows, see Section 5.2.2, “Service Manager Installation on Windows”, for Mac OS X see Section 5.2.3, “Service Manager Installation on Mac OS X”, and for Unix/Linux, see Section 5.2.4, “Service Manager Installation on Unix”.

5.2.1. 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 location of the MySQL server, the Tomcat server, and the connection properties for the MySQL server (Repository Configuration):

- **Tomcat Server Options**

  - Tomcat Server port: The default port that the Tomcat server uses when listening for connections. If you change this option, make the same change to the port number used to connect to the Service Manager. The default value is 18080.

    > **Note**
    >
    > If you do not have a web server running on port 80, you can use this 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.

- **MySQL Server selection**

  - You are asked whether you want to use your own MySQL Server to store repository data, or use the bundled MySQL Server. The MySQL server can be located on the same machine as the MySQL Enterprise Service Manager, or on a different machine. If you choose to use your own server, it must already be running when you start the installer. The requirements for the MySQL server are:

    - MySQL 5.1.43 or higher.

    - Partitioning enabled.

    - InnoDB Storage Engine enabled.

    Configure the MySQL server with suitable credentials to allow MySQL Enterprise Service Manager to create the database, tables and other settings. The user should have `ALL` privileges on the MySQL Enterprise Monitor repository, which is stored within the database `mem`, or you can provide full access to all tables. Also ensure you have the `GRANT OPTION`. For example, you might create a suitable user using the following `GRANT` statement:

    ```
    mysql> GRANT ALL PRIVILEGES ON *.* TO 'service_manager'@'localhost'
        -> IDENTIFIED BY PASSWORD 'PASSWORD' WITH GRANT OPTION
    ```

- **Repository Configuration (MySQL Server)**
  - Repository Username — if you are connecting to an existing MySQL database, this should be the username and password that you have configured with access to the MySQL Enterprise Monitor repository. If you install the bundled MySQL database, the credentials given are used as the username/password for the MySQL Enterprise Service Manager and are created automatically during installation. 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 connecting to the MySQL database. This should be set to a secure password so that the repository data is secure.

The information that you configure during installation is 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. Secure the installation directory and file, so that they can only be accessed by users who need to use the information.

5.2.2. Service Manager Installation on Windows

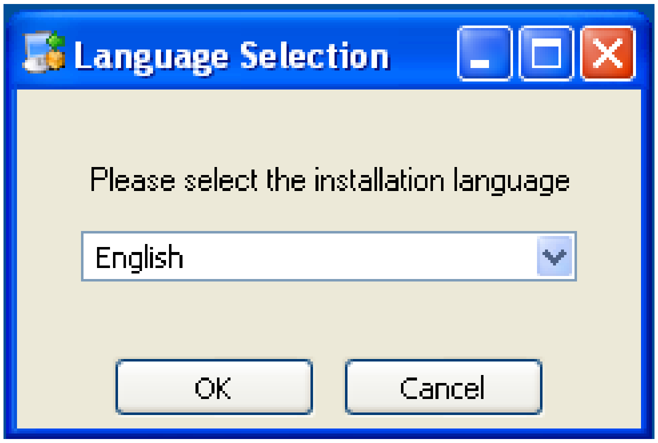
On Windows, the installation modes are `win32` and `unattended` only. `unattended` mode is especially useful if you are doing multiple installations. For more information on this topic see Section 5.4, “Unattended Installation”.

> **Note**
>
> In order 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 5.1. MySQL Enterprise Monitor: Installing Monitor on Windows: Language Selection

   Language Selection

   Please select the installation language

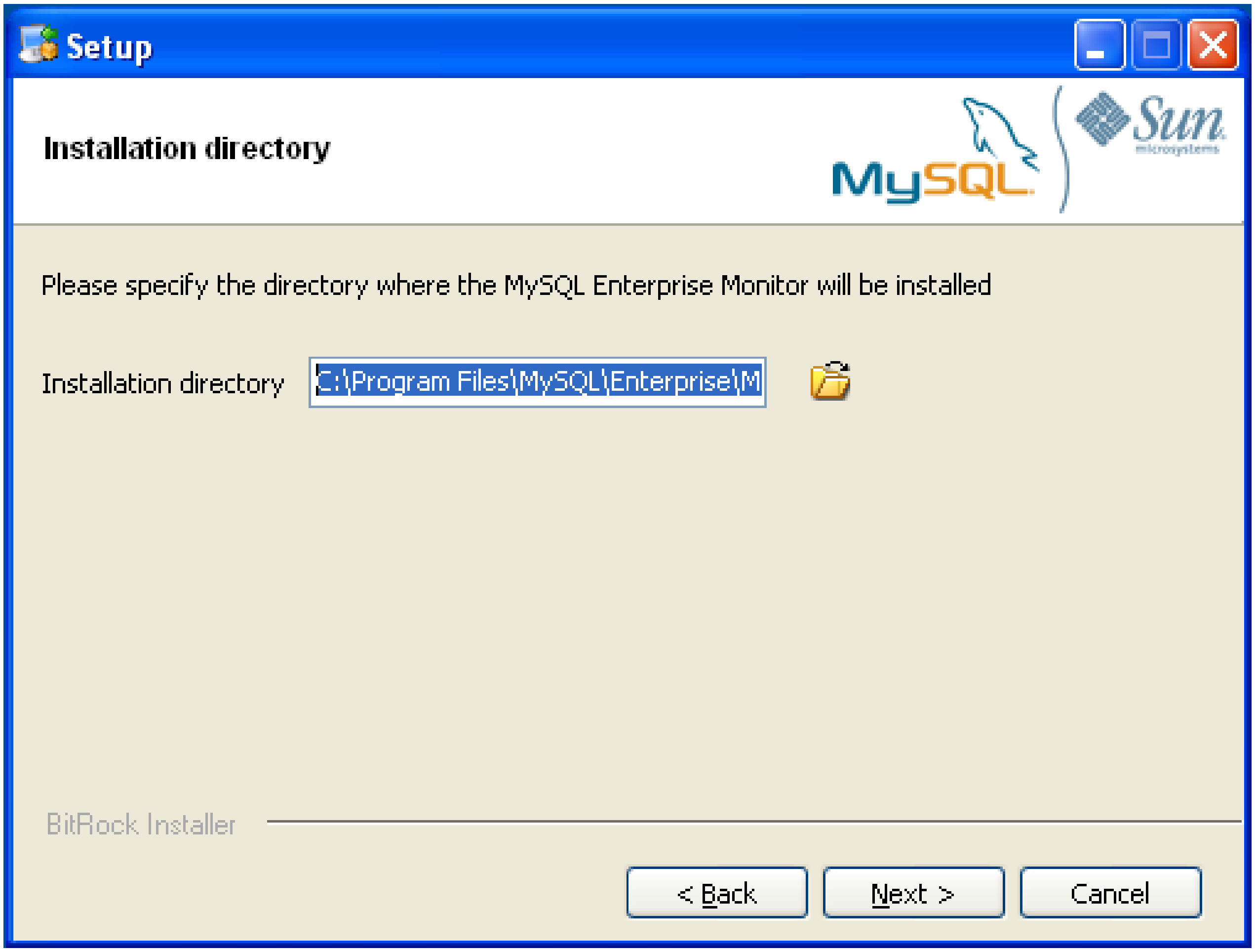
   English

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 Win-

dows 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 5.2. MySQL Enterprise Monitor: Installing Monitor on Windows: Installation Directory



Setup

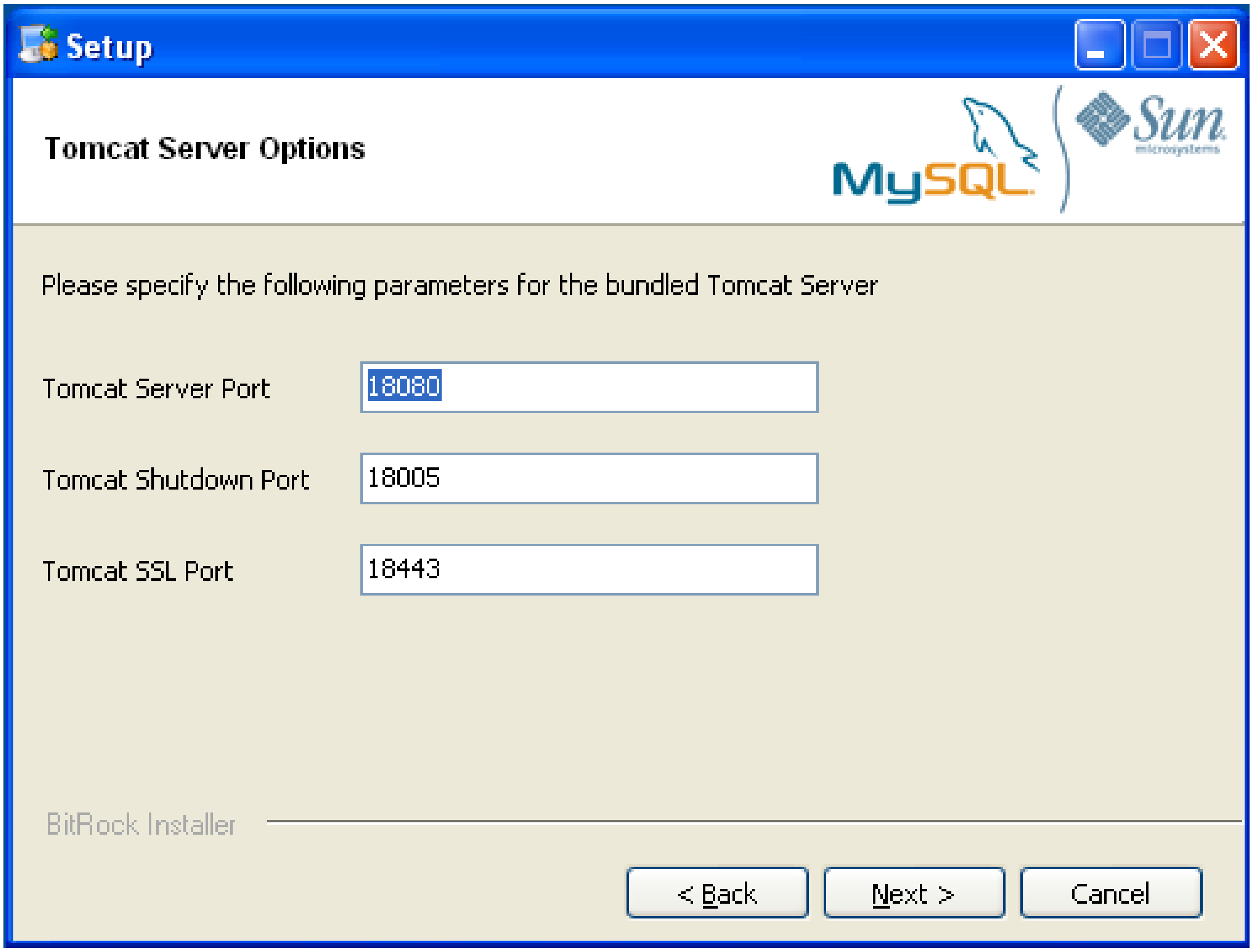
Installation directory

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 5.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

Figure 5.3. 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

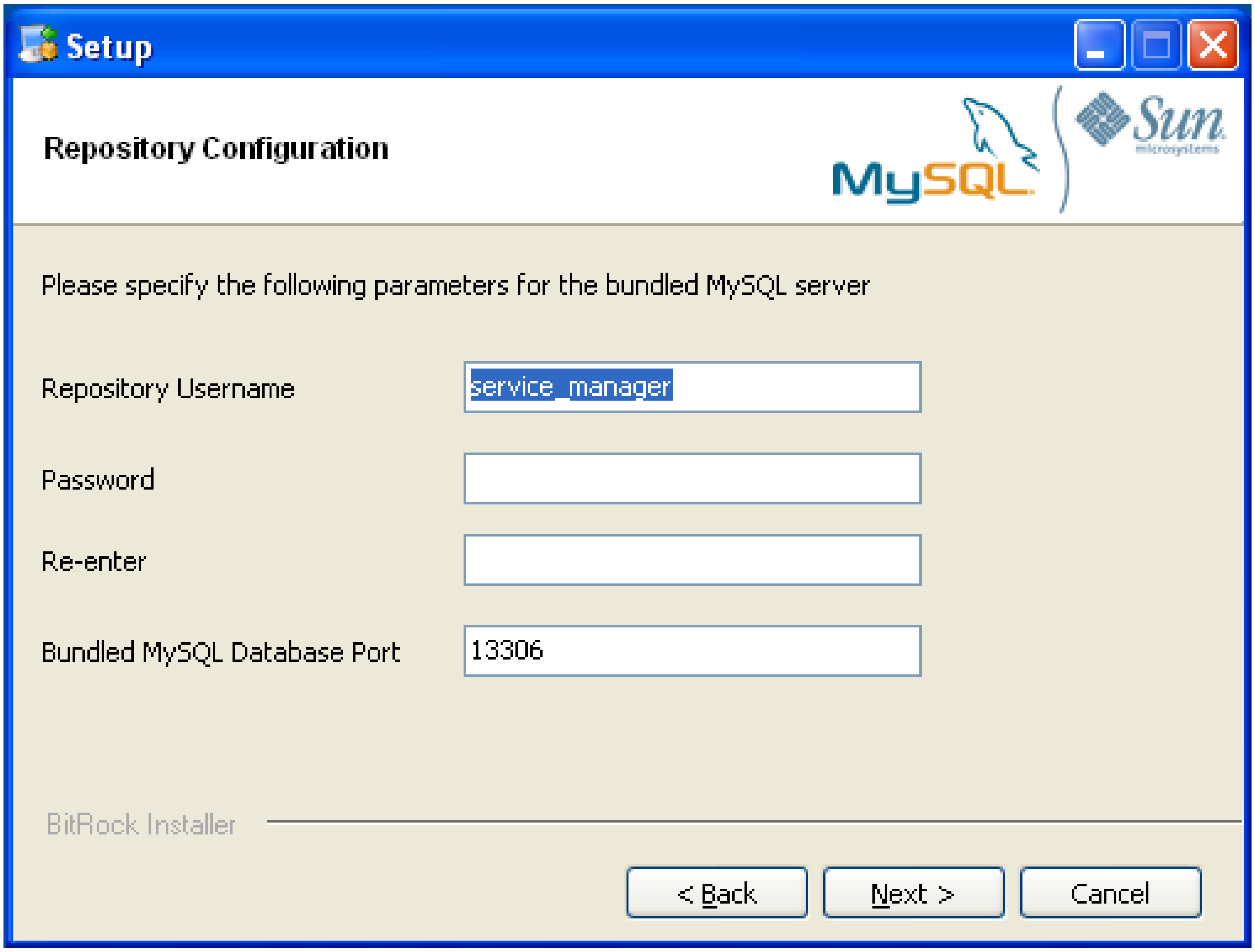
Tomcat SSL Port: 18443

5. Configure the repository settings, setting the user name, password and port for the bundled MySQL server that stores the information and statistics for your installation. For more information, see Section 5.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

   Note

   If the Windows firewall is enabled, unblock ports for Apache/Tomcat and the MySQL server.

Figure 5.4. 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 receive information and a warning about the configuration options and how they are stored in the `configuration_report.txt` file, and its location. Record the full path to this file for future reference. Click NEXT to continue.

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

8. Once the installation is completed, you receive 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, login to the Dashboard. You can do this automatically by leaving the box checked on the final window before clicking FINISH. To skip the Dashboard at this time, uncheck the box and click FINISH.

For instructions on starting the MySQL Enterprise Monitor services under Windows, see Section 5.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows”.

5.2.3. 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 5.4, “Unattended Installation”. The `text` mode installation for Mac OS X is identical to `text` installation under Unix. For `text` mode installation instructions see Section 5.2.4, “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.

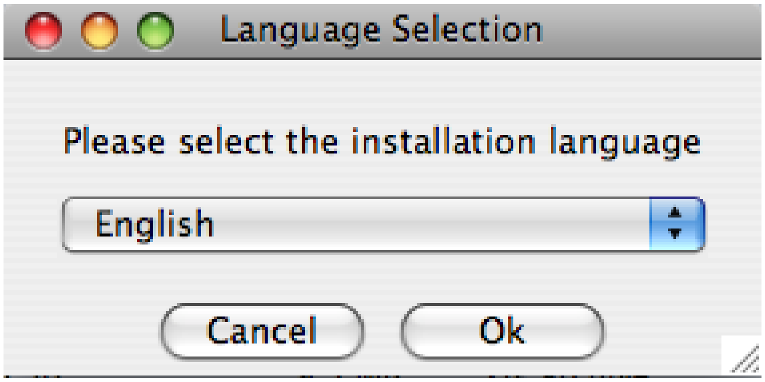
For reasons of backwards compatibility, Mac OS X is usually installed with multiple versions of Java. When installing in osx mode, version 1.5.0_7 must be the default version. Upon installation, Java 1.5.0_7 sets itself as the default so this is usually not a problem.

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 5.2.4, “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 5.5. MySQL Enterprise Monitor: Installing Monitor on OS X: Language Selection



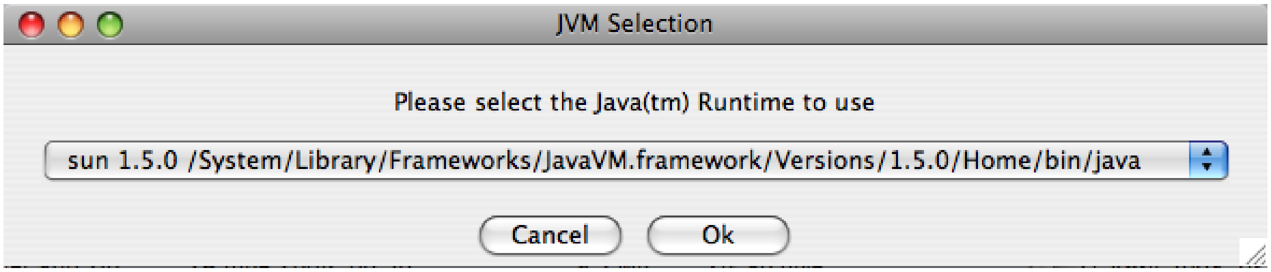
Language Selection

Please select the installation language

English

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

Figure 5.6. MySQL Enterprise Monitor: Installing Monitor on OS X: Java Selection



JVM Selection

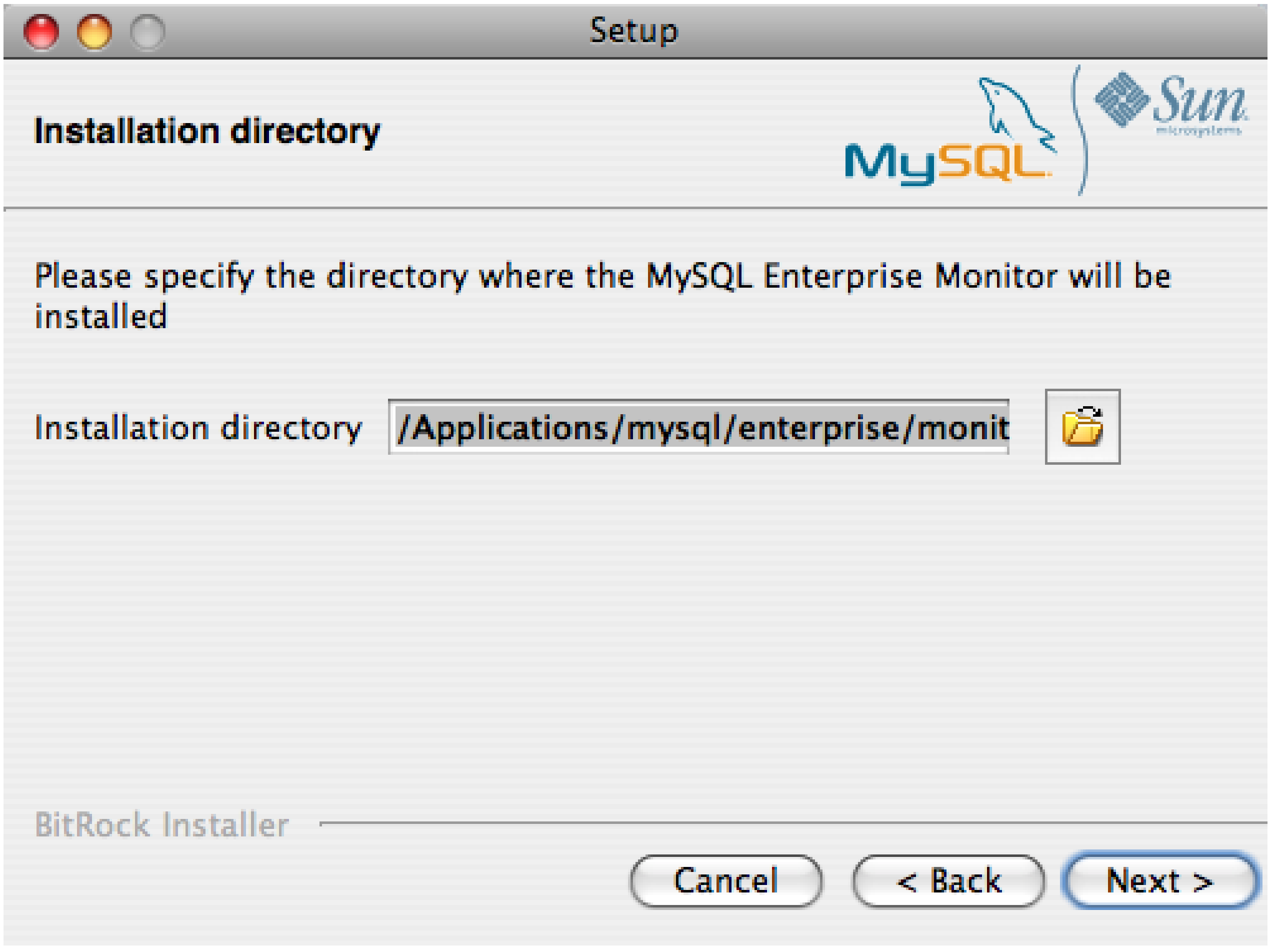
Please select the Java(tm) Runtime to use

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 continue.

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 5.7. MySQL Enterprise Monitor: Installing Monitor on OS X: Installation Directory



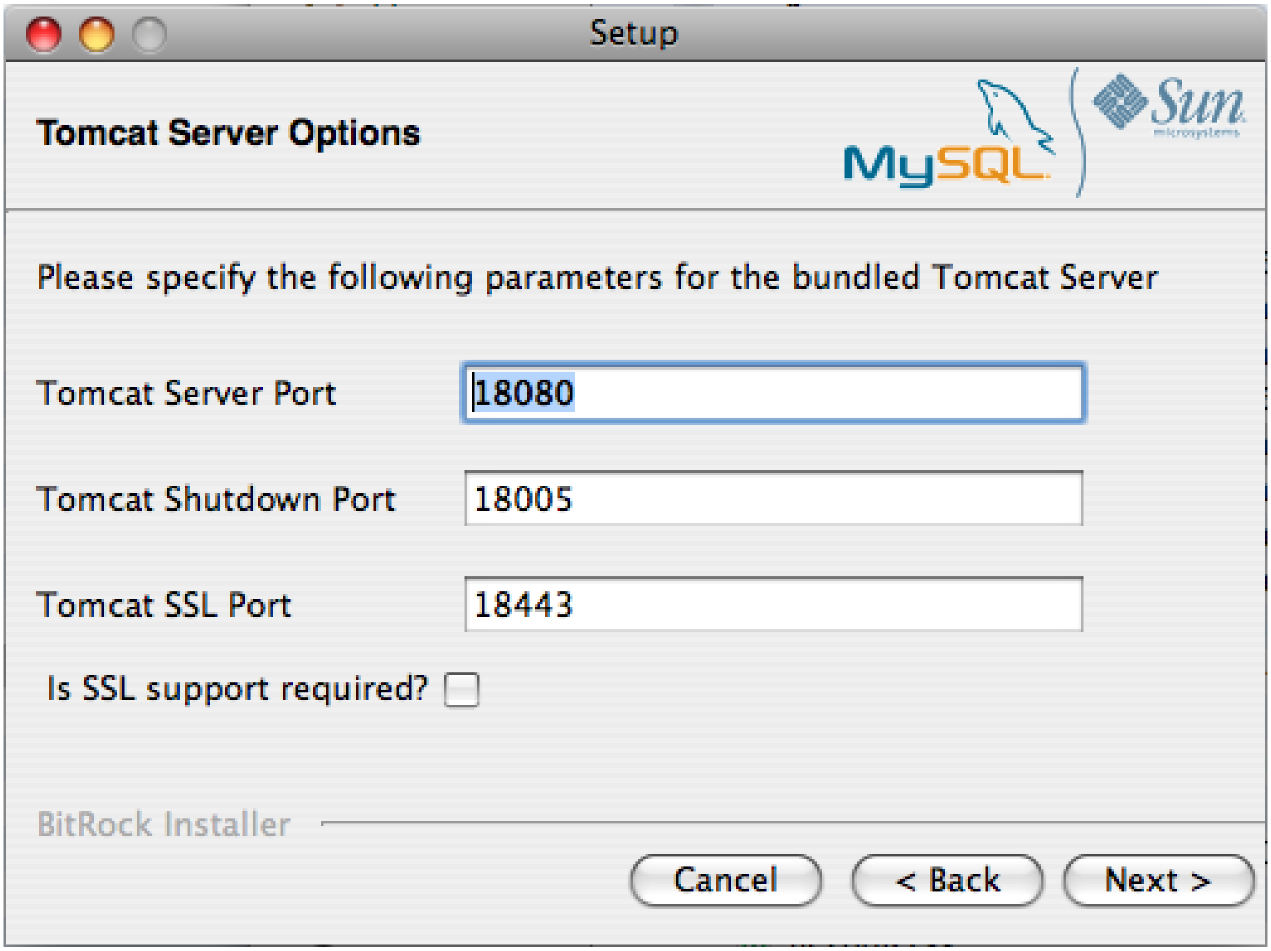
Installation directory

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 5.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

Figure 5.8. 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

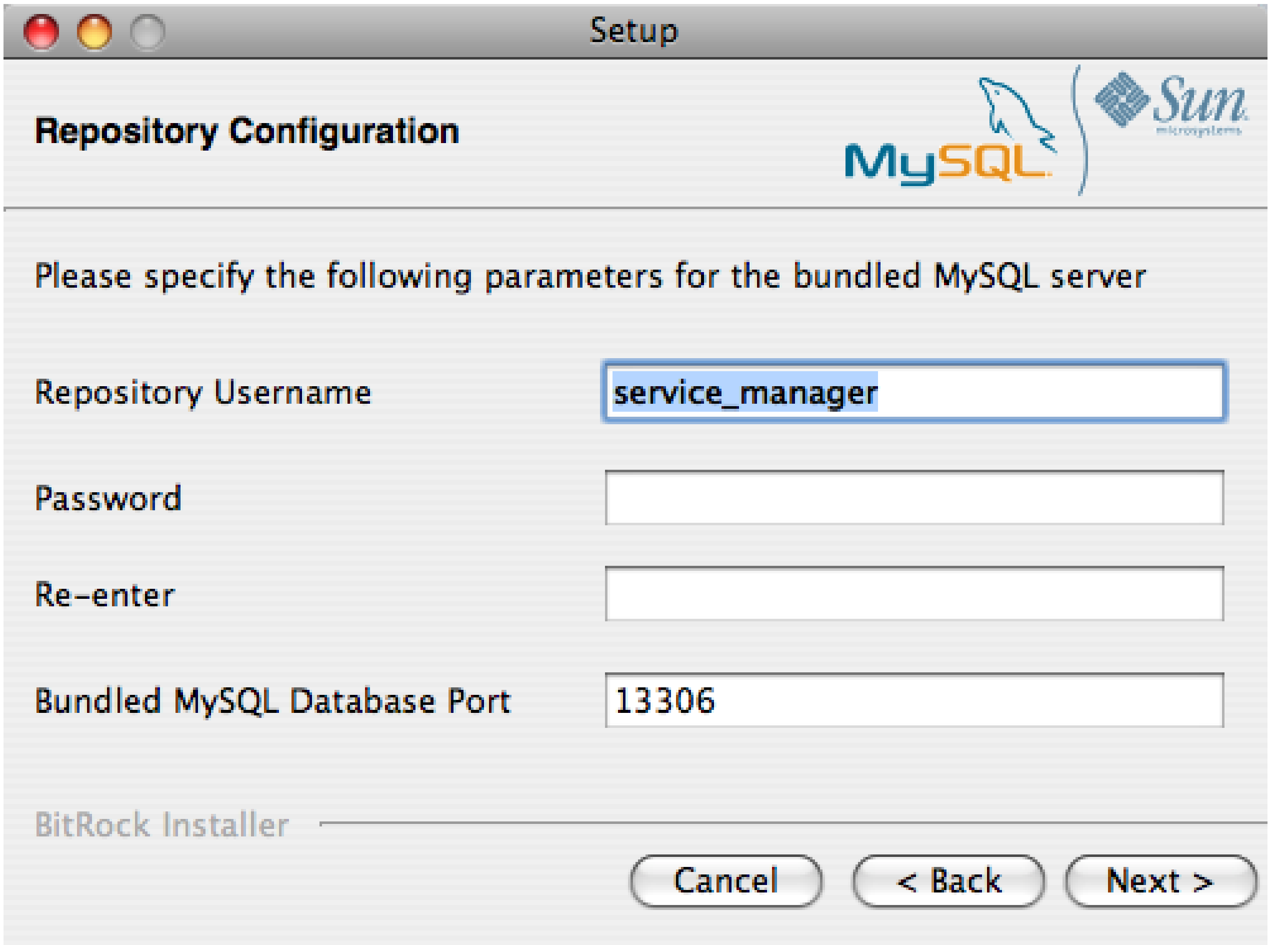
Tomcat Shutdown Port: 18005

Tomcat SSL Port: 18443

Is SSL support required? □

6. Configure the repository settings, setting the user name, password and port for the bundled MySQL server that stores the information and statistics for your installation. For more information, see Section 5.2.1, “Service Manager Installation Common Parameters”. Click NEXT to continue.

Figure 5.9. 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 receive information and a warning about the configuration options and how they are stored in the `configuration_report.txt` file, and its location. Record the full path to this file for future reference. Click NEXT to continue.

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

9. Once the installation is completed, you receive 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, login to the Dashboard. You can do this automatically by leaving the box checked on the final window before clicking FINISH. To skip the Dashboard at this time, uncheck the box and click FINISH.

Your installation is now complete. To continue with the configuration of MySQL Enterprise Service Manager, see Section 5.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

5.2.4. 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`), login as `root`. An unprivileged user installs to the `/home/user_name/mysql/enterprise/monitor/` directory.

What follows describes installation from the command line. You can install the Service Manager graphically by running the installer from within a windows manager. In both cases, the steps are identical. You can 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 5.4, "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 are 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 to install MySQL Enterprise Service Manager. 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 is 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 5.2.1, "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]:
   ```

   Choose if SSL support is required. When your agents and the monitor communicate with each other using SSL, the data is exchanged in encrypted form and you can monitor servers securely, or monitor agents over a public connection.

   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 5.2.1, "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 receive the location of the file that contains a copy of all of the settings. 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. 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 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 can read the `Readme` file that 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 does not start up automatically if you perform a `text` mode installation. For more information on starting and stopping MySQL Enterprise Service Manager, see Section 5.2.6, “Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X”.

5.2.5. 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 5.2.7, “MySQL Enterprise Service Manager Configuration Settings and Advisor Installation”.

5.2.6. 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 are 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.

5.2.7. 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).

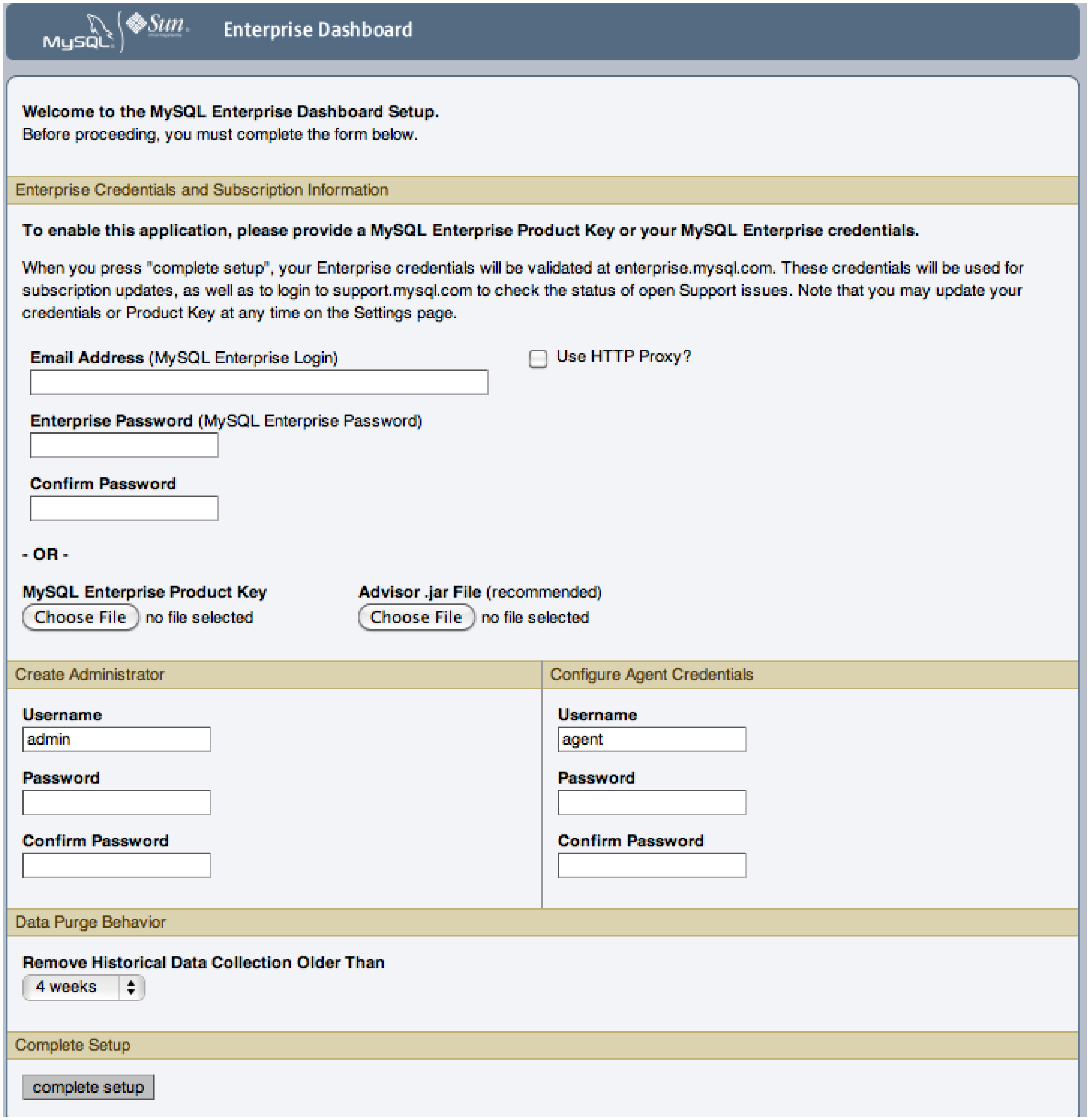
Otherwise, you can view the dashboard by typing `http://localhost:18080/Auth.action` into the address bar of your web browser. If you are unsure of the host name and port to use, check the `configuration_report.txt` file.

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`.

5.2.7.1. 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 5.10. MySQL Enterprise Monitor: Initial Dashboard Log-In



Enterprise Dashboard

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 be used for subscription updates, as well as to login to support.mysql.com to check the status of open Support issues. Note that you may update your credentials or Product Key at any time on the Settings page.

Email Address (MySQL Enterprise Login)

□ Use HTTP Proxy?

Enterprise Password (MySQL Enterprise Password)

Confirm Password

- OR -

MySQL Enterprise Product Key
Choose File no file selected

Advisor .jar File (recommended)
Choose File no file selected

Create Administrator

Username
admin

Password

Confirm Password

Configure Agent Credentials

Username
agent

Password

Confirm Password

Data Purge Behavior

Remove Historical Data Collection Older Than
4 weeks

Complete Setup

complete setup

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 is named, `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 5.2.7.3, "Installing, Updating or Upgrading Advisors After Initial Login". If the product key that you provide is invalid, a notification appears and you cannot 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 5.1.3, "Users Created on First Log-in". Record 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 5.1.3, "Users Created on First Log-in". The agent must log in to report its findings. Record 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 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 5.2.7.3, "Installing, Updating or Upgrading Advisors After Initial Login".

5.2.7.2. 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 2.8.2, "User Preferences".
>
> The locale chosen determines the user's default language when logging in to the Dashboard. This overrides the default browser settings whenever this specific user logs in.

After specifying your time zone and locale, the dashboard opens on the **Monitor** tab. For a detailed examination of the **Monitor** Screen, see Section 2.1, "MySQL Enterprise Dashboard".

5.2.7.3. Installing, Updating or Upgrading Advisors After Initial Login

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. To install the advisors in this fashion, specify your MySQL Enterprise credentials. Find instructions for doing this in Section 2.8.1, "Global Settings".

If you do not allow Internet access from the dashboard, install the Advisors from a local file. You need an advisor file named `Ad-`

visorScript-*version*.jar. If you do not 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.

5.3. Monitor Agent Installation

A MySQL Enterprise Agent monitors a MySQL server and sends data to the Advisors. The data is 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.

5.3.1. Creating a MySQL User Account for the Monitor Agent

The MySQL Enterprise Agent requires a user configured within each MySQL instance that is being monitored with suitable privileges to collect information about the server, including variable names, replication, and storage engine status information. The agent installer can create a suitable user for you during installation if you supply the installer with a user/password for a user with suitable credentials (such as the root user). The account information is used only during the installation to perform the user creation. If you want to use this option, you can skip this section.

If you do not supply the root user information to the installer, you can create a user manually within your MySQL server and provide these credentials as the agent user/password combination during installation. The privileges required for this user account vary depending on the information you gather using the MySQL Enterprise Agent. The following privileges allow the Monitor Agent to perform its assigned duties without limitation:

- SHOW DATABASES: The MySQL Enterprise Agent can gather inventory about the monitored MySQL server.
- REPLICATION CLIENT: The MySQL Enterprise Agent can gather Replication master/slave status data. This privilege is only needed if the MySQL Replication Advisor Rules are employed.
- SELECT: The MySQL Enterprise Agent can collect statistics for table objects.
- SUPER: The MySQL Enterprise Agent can execute SHOW ENGINE INNODB STATUS to collect data about InnoDB tables. This privilege is also required to obtain replication information using SHOW MASTER STATUS, and to temporarily switch off replication when populating the mysql.inventory table used to identify the MySQL instance.
- 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: The MySQL Enterprise Agent can 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, which it sends along with other information to MySQL Enterprise Service Manager.

For example, the following GRANT statement gives 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 might 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, 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 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, a warning containing this information is 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 is *`localhost`*. However, this changes 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 5.3.7, “Configuring an Agent to Monitor a Remote MySQL Server”.

5.3.2. Installing the Agent on Microsoft Windows

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

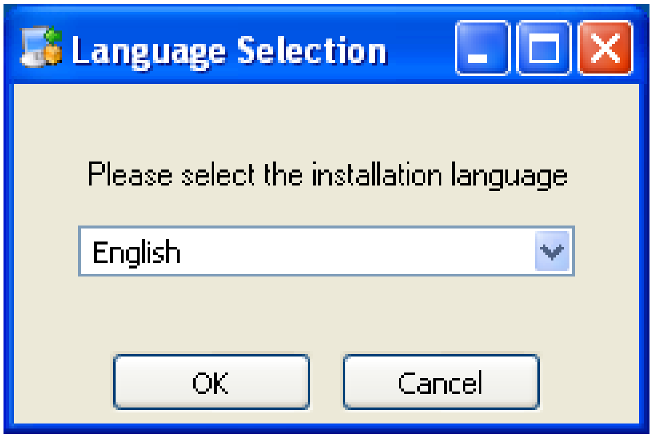
> **Note**
>
> In order to install the agent as a Windows service, you must do the installation as a privileged user.
>
> On Windows Vista or later, if user account control is on, an operating system dialog box requests confirmation of the installation.

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 5.4, “Unattended Installation”.

> **Note**
>
> If you want to install multiple agents on the same machine, you must use the `servicename` option to the installer prevent installing an agent with the same service name. For more information, see `installer_servicename` If the installer identifies an existing installation and you do not specify an alternative service name, the installer terminates.

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

   Figure 5.11. MySQL Enterprise Monitor: Installing Agent on Windows: Language Selection



Language Selection

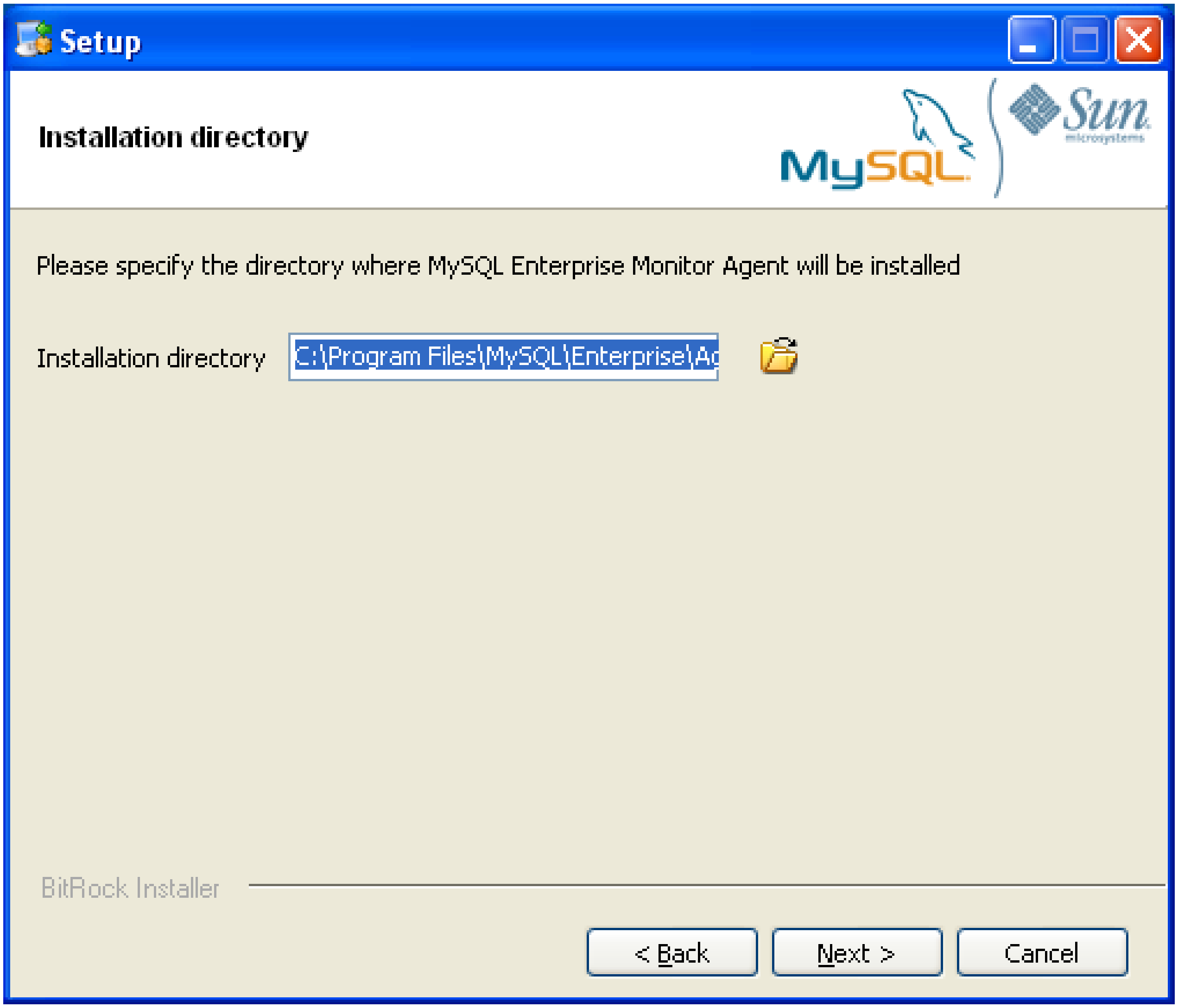
Please select the installation language

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 5.12. MySQL Enterprise Monitor: Installing Agent on Windows: Installation Directory



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

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

4. You need to choose the operations that you want this agent to perform. You choose from one or more of the following areas of functionality:

   - Enable monitoring a MySQL Server: This is the main operation of monitoring the MySQL server in terms of performance, security and availability.

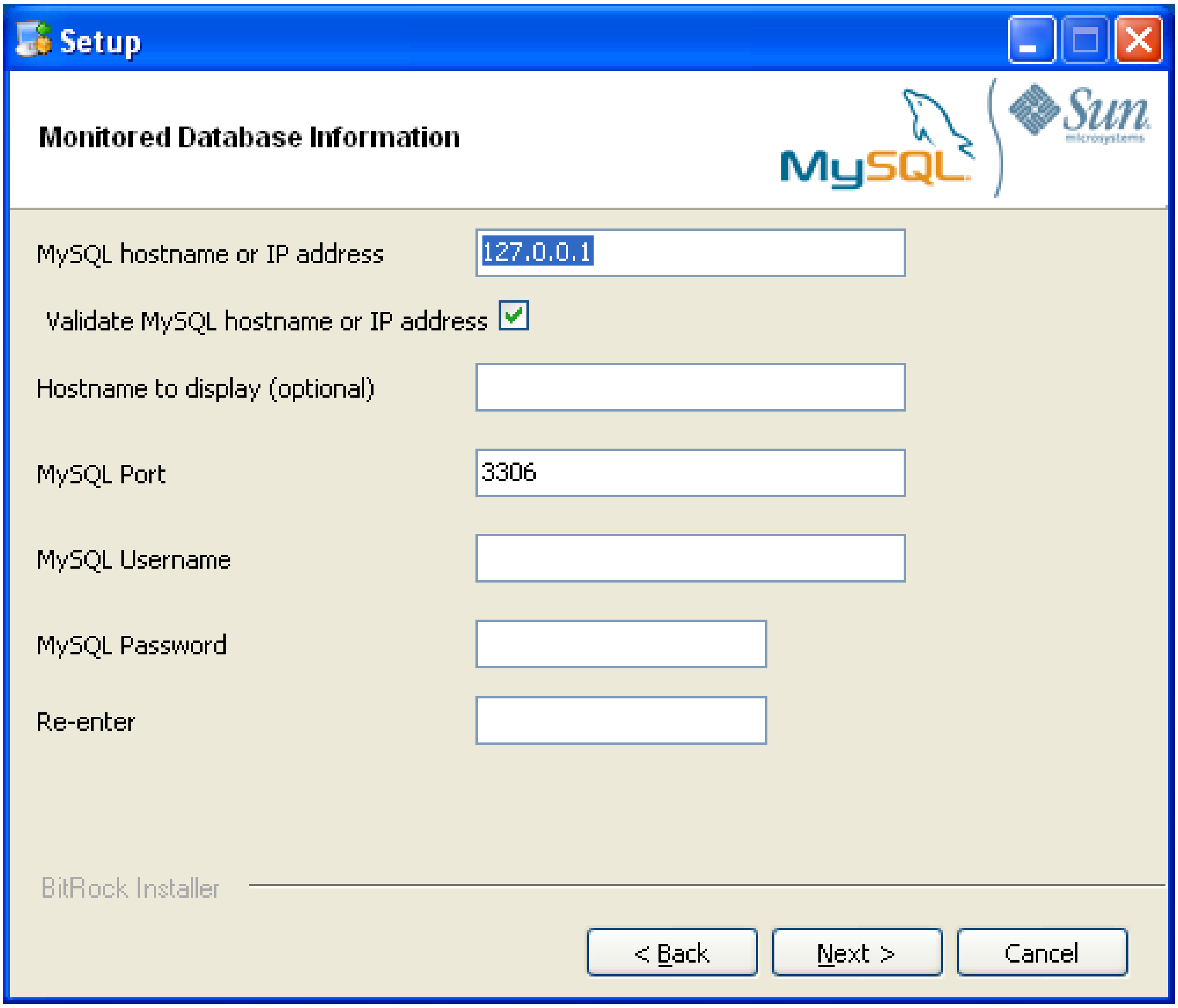
   - Enable the MySQL Enterprise Agent Proxy Service: If you want to study your query analyzer information by sending your queries through the MySQL Enterprise Agent Proxy Service you must switch this option on.

5. Specify the information about the MySQL server that you want to monitor. Enter the IP address or host name of the host you want to monitor, and the port, user name and password that you use to connect to the MySQL server. To confirm that the MySQL server is currently reachable using the information, select the **VALIDATE MYSQL HOST NAME OR IP ADDRESS** checkbox.

   > **Note**
   >
   > Currently, on Windows, the monitor agent only includes support for connecting to the server to be monitored via 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`, the server only listens for connections on the specified IP address (of the MySQL server). If the monitor agent uses TCP/IP networking and the default address of 127.0.0.1, it cannot connect to the server to be monitored. Also, if “localhost” is specified as the

host name during agent configuration, a connection cannot be established, as the server listens for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

Figure 5.13. MySQL Enterprise Monitor: Installing Agent on Windows: Monitored Database Information



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:

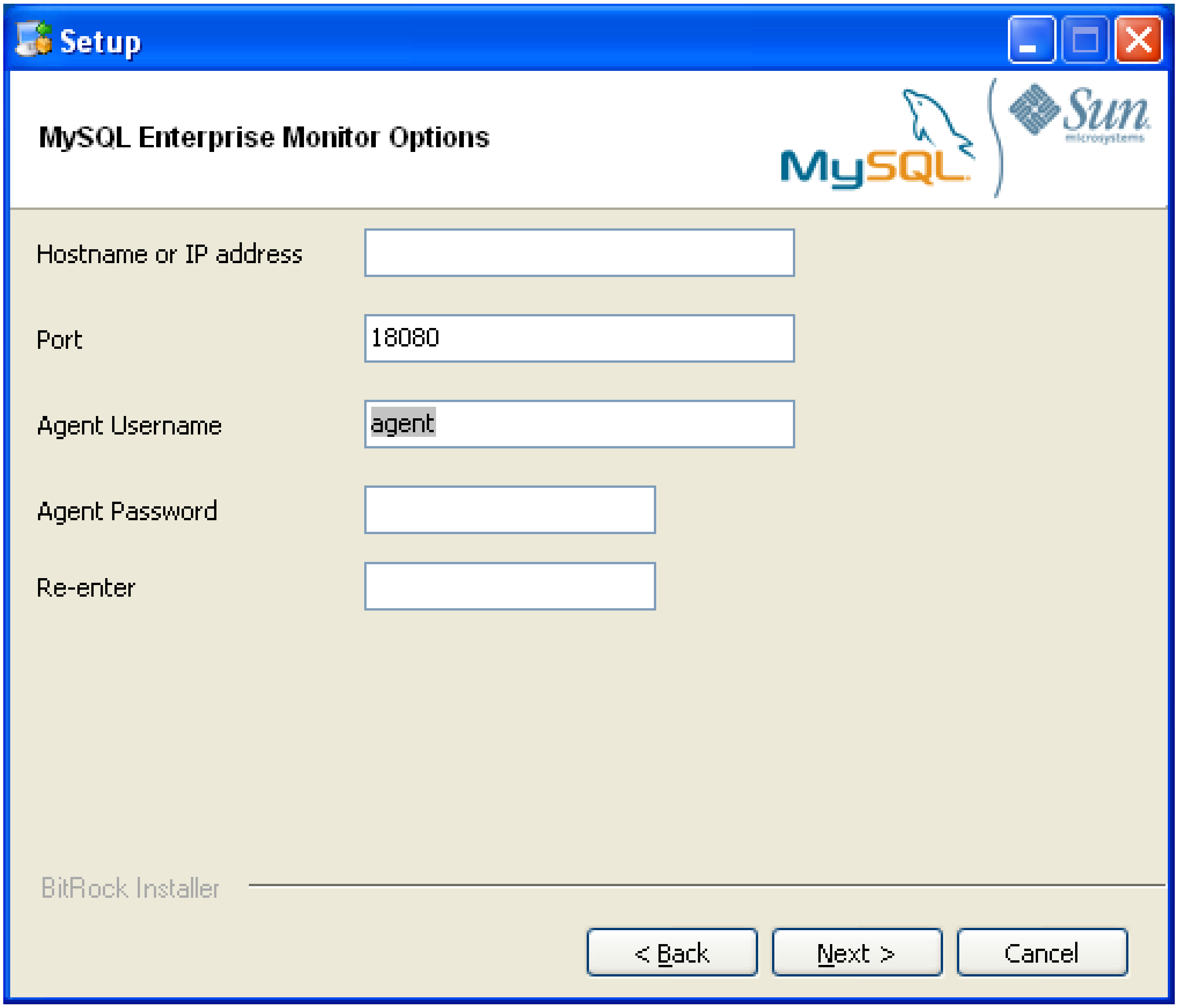
Click NEXT to continue the installation.

6. The installer can create the MySQL Enterprise Agent user on the MySQL server that is required to connect to the server and obtain statistical information. You can elect to create an account automatically using the installer, or you can use an account that you have already created. If you want the installer to create the user, enter the credentials of a user with the correct abilities to create the user and grant the created user the right abilities to monitor the server. By default, the installer requests the rights of a `root` user to perform this operation.

7. 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 5.14. MySQL Enterprise Monitor: Installing Agent on Windows: MySQL Enterprise Service

Manager Options



Hostname or IP address

Port: 18080

Agent Username: agent

Agent Password

Re-enter

8. If you want to use Query Analyzer with the MySQL Enterprise Agent Proxy Service, then you need to enable the MySQL Enterprise Agent Proxy service and configure the port and other information. If you disable the Proxy during installation, enable it later before you can use Query Analyzer. For more information on Query Analyzer, see Section 2.6, “The **QUERY ANALYZER** Tab”.

   When Proxy is enabled, MySQL Enterprise 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 Agent listens for connections.

   The default port is 6446.

9. A Configuration Report summarizes the information that you 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.

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

11. Once the agent is installed, you get a confirmation message. Click NEXT to finalize the installation.

12. You can start the MySQL Enterprise Agent automatically now that the installation has been completed. To start the agent immedi-

ately, leave the checkbox selected. To start the agent separately, uncheck the checkbox. Click FINISH to exit the installation.

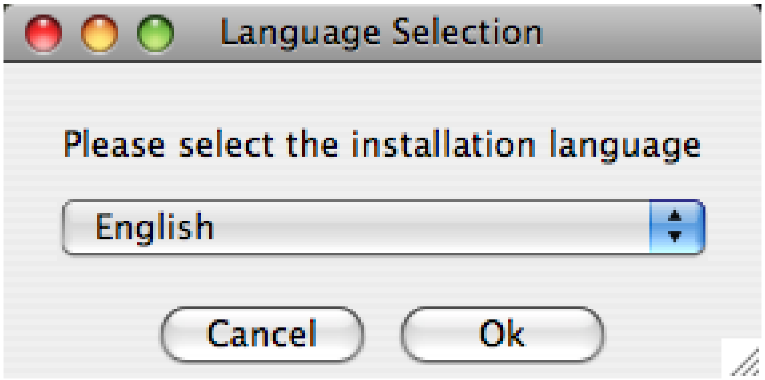
Once the Monitor Agent is installed, it needs to be started. For information on how to start and stop the Agent, see Section 5.3.5.1, “Starting/Stopping the Agent on Windows”.

5.3.3. Installing the Agent on Mac OS X

To install the MySQL Enterprise 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 Agent installation. Click OK to continue installation.

   Figure 5.15. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Language Selection

   Language Selection

   Please select the installation language

   English

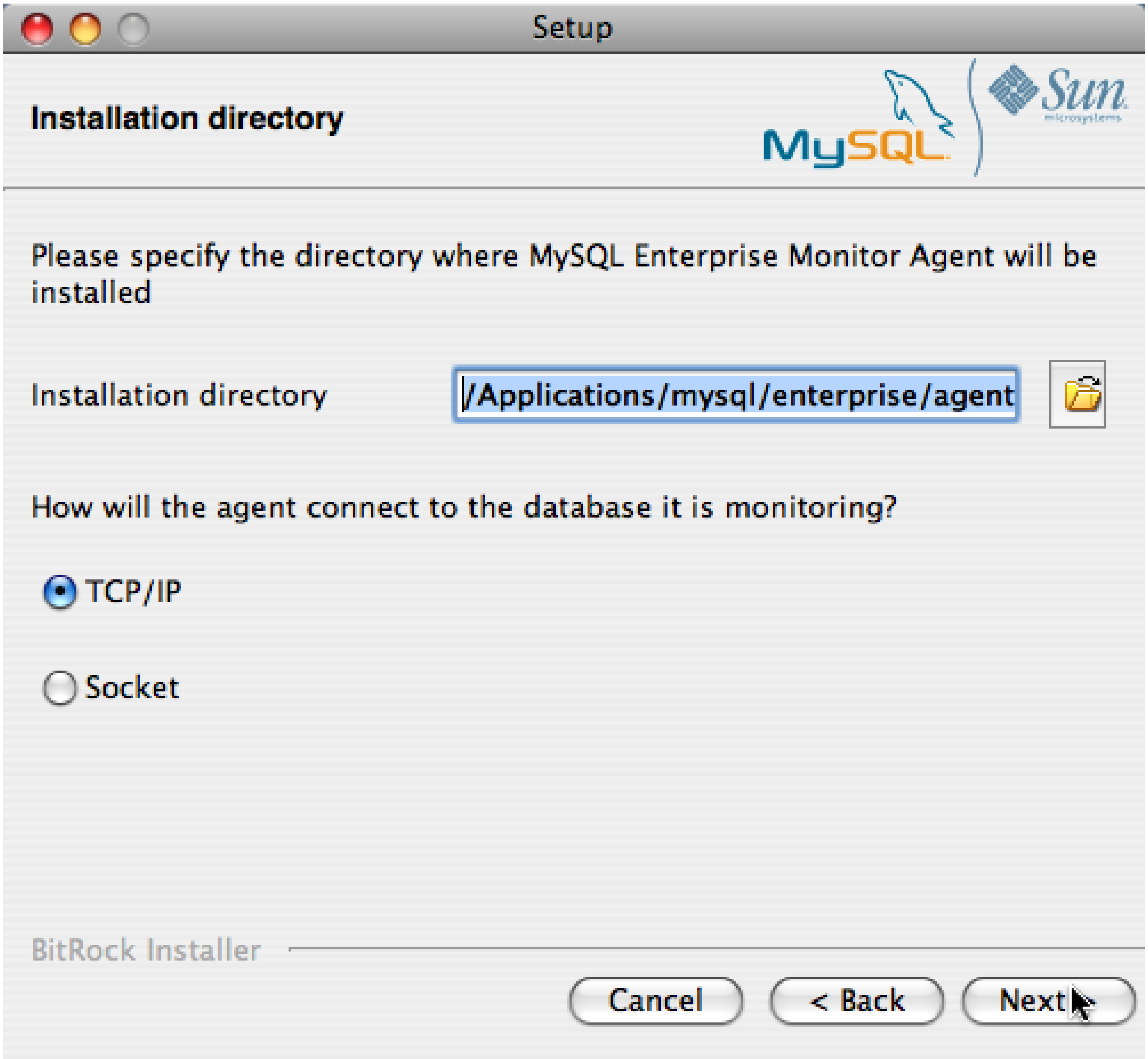
2. Click NEXT to start the installation process.

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.

   You also need to select the method that the agent uses 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 via the MySQL socket by default on Unix, if the hostname “localhost” is specified.
   >
   > If the MySQL server to monitor is started with the `--skip-networking` command option, you cannot connect to it via TCP/IP, as the server does not listen for TCP/IP connections. In this case, configure the monitor agent 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 C.2.2, “MySQL Server (`agent-instance.ini`) Configuration”.
   >
   > If the MySQL server to be monitored is started using the command option `--bind-address`, the server only listens for connections on the IP address specified, that is, the IP address of the MySQL server. If the monitor agent is started using TCP/IP networking and the default address of 127.0.0.1, it cannot connect to the server to be monitored. Also, if “localhost” is specified as the host name during agent configuration, a connection cannot be established, as the server listens for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

   Figure 5.16. MySQL Enterprise Monitor: Installing Agent on Mac OS X: 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 choose the operations that you want this agent to perform. You choose from one or more of the following areas of functionality:

   - Enable monitoring a MySQL Server: This is the main operation of monitoring the MySQL server in terms of performance, security and availability.

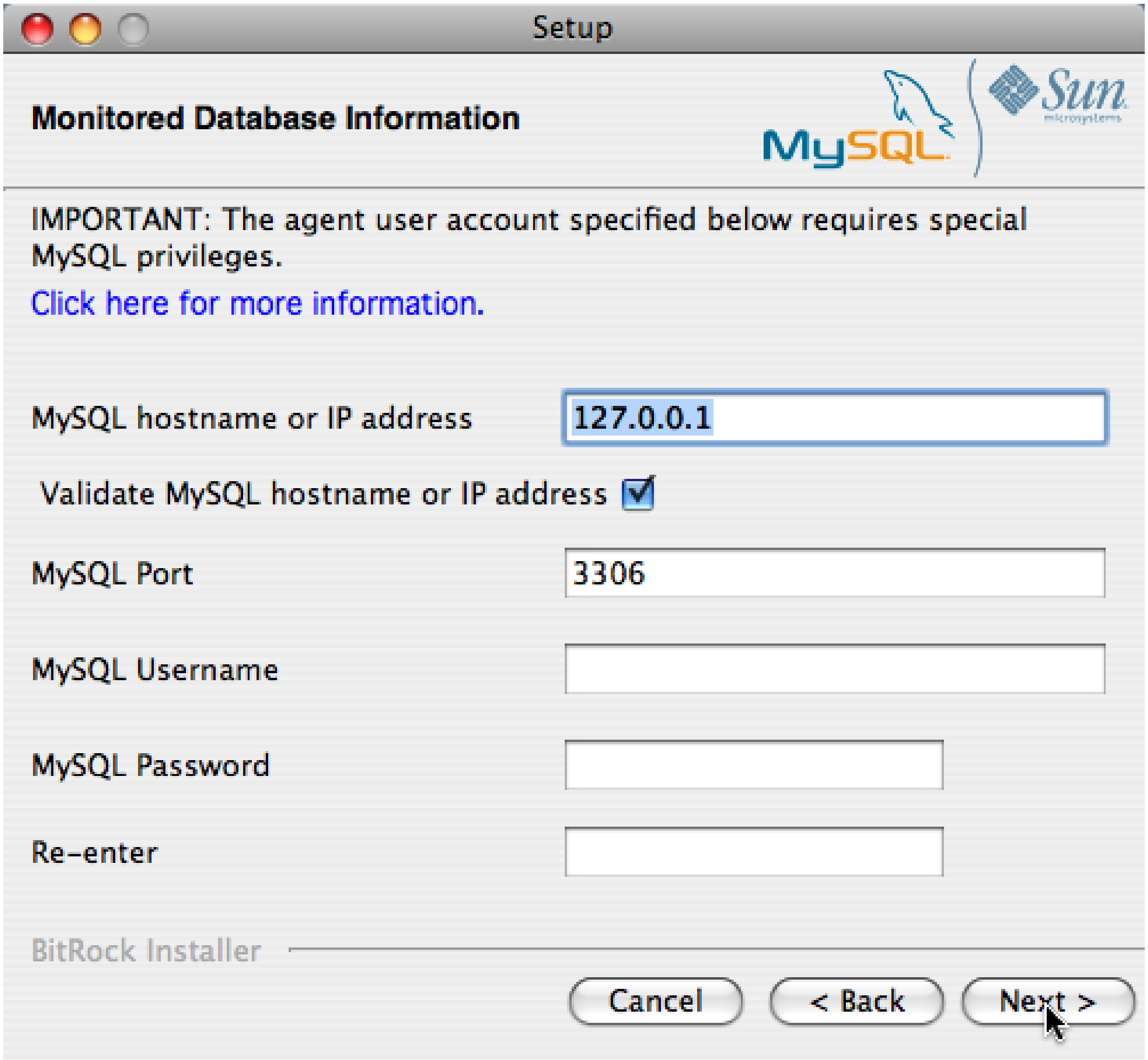
   - Enable the MySQL Enterprise Agent Proxy Service: If you want to study your query analyzer information by sending your queries through the MySQL Enterprise Agent Proxy Service you must switch this option on.

5. Specify the information about the MySQL server that you want to monitor. The configuration information you enter depends on the connection method selected in the previous screen.

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

Figure 5.17. 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

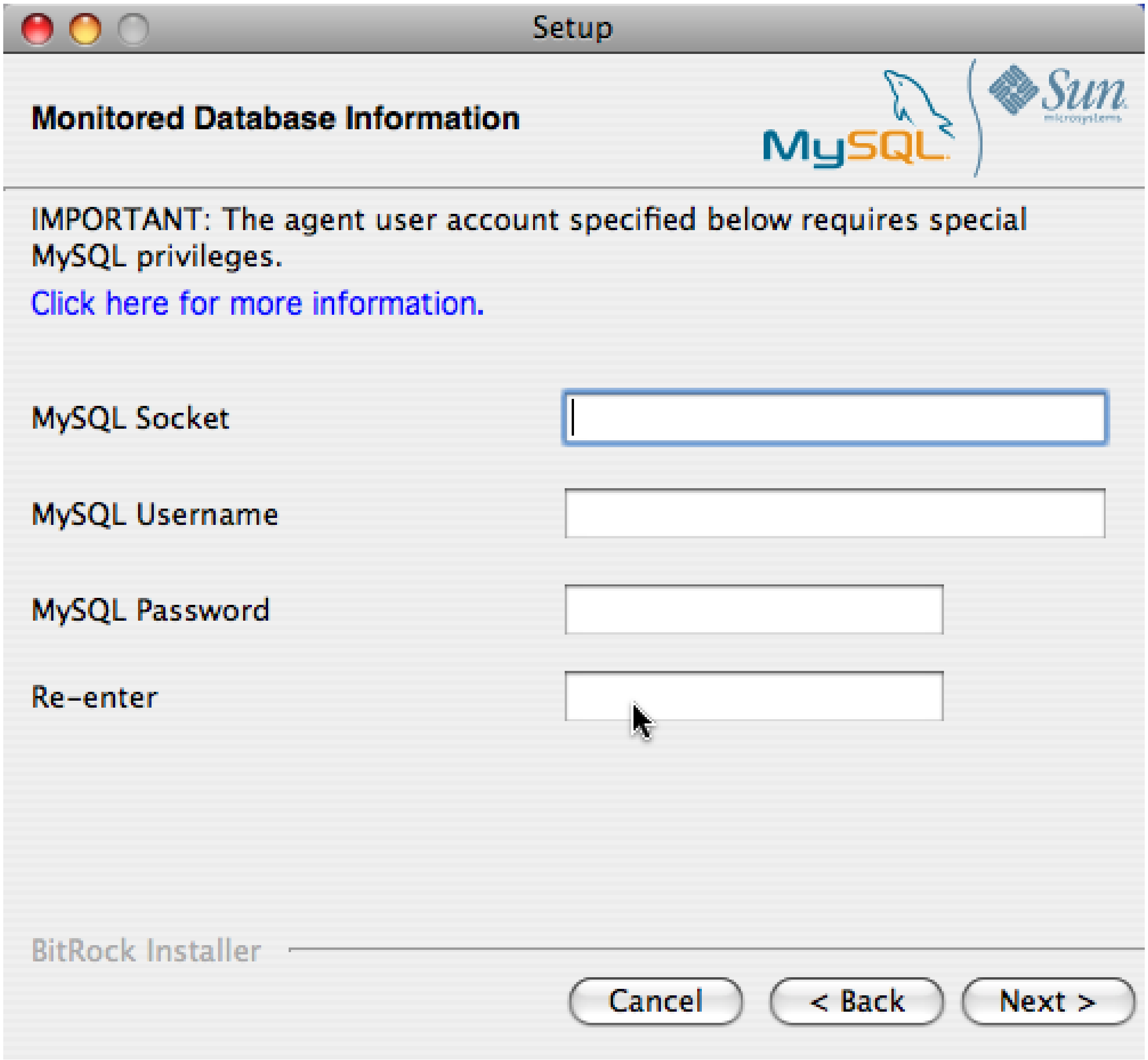
MySQL Username:

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 to authenticate with the server. Typical values include `/tmp/mysql.sock` and `/var/mysql/mysql.sock`.

Figure 5.18. 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

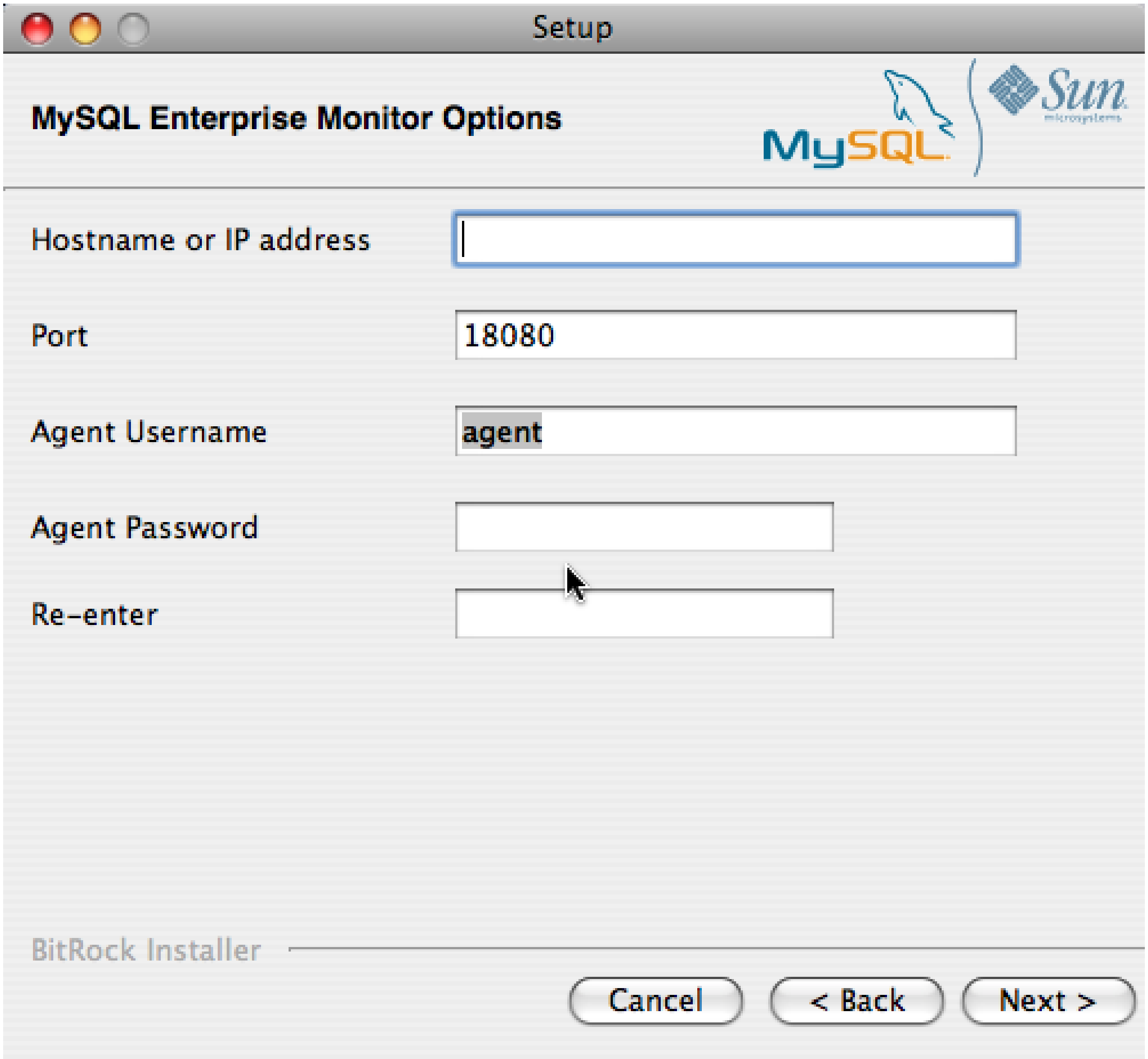
Re-enter

Click NEXT to continue the installation.

6. The installer can create the MySQL Enterprise Agent user on the MySQL server that is required to connect to the server and obtain statistical information. You can elect either to create an account automatically using the installer, or you can use an account that you have already created. If you want the installer to create the user, you must enter the credentials of a user with the correct abilities to create the user and grant the created user the right abilities to monitor the server. By default, the installer requests the rights of a `root` user to perform this operation.

7. 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 5.19. 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

8. If you want to use Query Analyzer with the MySQL Enterprise Agent Proxy Service, then you need to enable the MySQL Enterprise Agent Proxy service and configure the port and other information. If you disable the Proxy during installation, enable it later before you can use Query Analyzer. For more information on Query Analyzer, see Section 2.6, "The **QUERY ANALYZER** Tab".

   When Proxy is enabled, MySQL Enterprise 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 Agent listens for connections.

   The default port is 6446.

9. A Configuration Report summarizes the information you 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.

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

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

12. You can start the MySQL Enterprise Agent automatically now the installation has been completed. To allow the agent to be star-

ted, 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 5.3.5.2, “Starting/Stopping the Agent on Mac OS X”.

5.3.4. Installing the Monitor Agent on Unix

As a prerequisite for installing the MySQL Enterprise 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 does 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 5.4, “Unattended Installation”.

Note

If you want to install multiple agents on the same machine, you must use the `servicename` option to the installer prevent installing an agent with the same service name. For more information, see `installer_servicename`

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. If an existing version of the MySQL Enterprise Agent is installed, provide an alternative name for the installation. This will be used as the name of the file created in `/etc/init.d` used to start the agent. You can enter a different name to perform multiple installations.

4. Specify the MySQL server to monitor. First, specify whether 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] :
   ```

5. You need to choose the operations that you want this agent to perform. You choose from one or more of the following areas of functionality:

   - Enable monitoring a MySQL Server: This is the main operation of monitoring the MySQL server in terms of performance, security and availability.

   - Enable the MySQL Enterprise Agent Proxy Service: If you want to study your query analyzer information by sending your queries through the MySQL Enterprise Agent Proxy Service you must switch this option on.

6. If you chose to monitor a MySQL server, provide the connection options for communicating with the server.

   If you selected TCP/IP, 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 selected Socket, 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 via the MySQL socket by default on Unix, if the hostname “localhost” is specified.
>
> If the MySQL server to monitor is started with the `--skip-networking` command option, you cannot connect to it via TCP/IP, as the server does not listen for TCP/IP connections. In this case, configure the monitor agent 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, refer to Section C.2.2, “MySQL Server (`agent-instance.ini`) Configuration”.
>
> If the MySQL server to be monitored is started using the command option `--bind-address`, the server only listens for connections on the IP address specified, that is, the IP address of the MySQL server. If the monitor agent is started using TCP/IP networking and the default address of 127.0.0.1, it cannot connect to the server to be monitored. Also, if “localhost” is specified as the host name during agent configuration, a connection cannot be established, as the server listens for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

7. Specify the user credentials for the MySQL server to monitor:

```
MySQL Username []: service_agent

MySQL Password :

Re-enter :
```

8. The installer can create the MySQL Enterprise Agent user on the MySQL server that is required to connect to the server and obtain statistical information. You can elect either to create an account automatically using the installer, or you can use an account that you have already created. If you want the installer to create the user, enter the credentials of a user with the correct abilities to create the user and grant the created user the right abilities to monitor the server. By default, the installer requests the rights of a `root` user to perform this operation.

9. If you selected to enable the MySQL Enterprise Agent Proxy Service for Query Analyzer, enter the configuration information. If you disable the Query Analyzer during installation, manually edit the configuration file to re-enable the Query Analyzer functionality later. If you enable Query Analyzer (Proxy), specify the port on which the agent listens 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 [6446]:

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 Section 2.6, “The **QUERY ANALYZER** Tab”.

10. 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 :

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

11. Before installation starts, you are provided with a summary of your installation settings:

```
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: 6446

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
```

12. The installer copies the necessary files and creates 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.
```

13. 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 5.3.5.3, “Starting/Stopping the Agent on Unix”.

5.3.5. Starting/Stopping the MySQL Enterprise Agent

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

> **Warning**
>
> If you use Query Analyzer, turning off the agent prevents your applications from communicating with the MySQL server. See Section 2.6, “The **QUERY ANALYZER** Tab”.

5.3.5.1. 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 Agent` entry. Simply select the `Start MySQL Enterprise 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 5.3.10, “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 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 C.2.2, “MySQL Server (`agent-instance.ini`) Configuration”, replace `MySQLEnterpriseMonitorAgent` with the appropriate agent name.

5.3.5.2. 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 run more than one agent on a specific machine, specify the path to the ini file when you stop the agent. Executing mysql-monitor-agent stop without an ini file only stops 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 5.3.10, “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 is only needed if the ini file is not installed at the default location, or you changed the name of the ini file. Use this option if you install 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 install the agent as root, on reboot the mysql-monitor-agent daemon starts up automatically. If you install the agent as an unprivileged user, manually start the agent on reboot or write a script to perform this task. Likewise, if you add an additional agent as described in Section C.2.2, “MySQL Server (agent-instance.ini) Configuration”, to start this agent on reboot, create a system initialization script appropriate to your operating system. To determine whether the agent is running, navigate to the init.d directory and issue the command **./mysql-monitor-agent status**.

> **Warning**
>
> To report its findings, the agent connects 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 5.3.10, “Troubleshooting the Agent”.

5.3.5.3. Starting/Stopping the Agent on Unix

When installation is finished, 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 is:

```
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, use kill to send a TERM signal to the running process:

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

If you run more than one agent on a specific machine, specify the path to the ini file when you stop the agent. Executing mysql-monitor-agent stop without an ini file only stops 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. 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 5.3.10, “Troubleshooting the Agent”.

Installation creates the directory /opt/mysql/enterprise/agent. The etc subdirectory contains the mysql-monitor-agent.ini file with the configuration settings. The log subdirectory holds log information.

To see all the command-line options available for the monitor agent, navigate to the /opt/mysql/enterprise/agent/etc/init.d directory and issue the command **mysql-monitor-agent** help, bringing up this help message:

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

You only need the ini-file-name option if the ini file is not installed at the default location, or you changed the name of the ini file. Use this option if you install 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 starts up automatically. If you installed the agent as an unprivileged user, manually start the agent on reboot or write a script to perform this task. Likewise, if you added an additional agent as described in Section C.2.2, “MySQL Server (agent-instance.ini) Configuration”, to start this agent on reboot, create a system initialization script appropriate to your operating system. To determine whether the agent is running, navigate to the init.d directory and issue the command **./mysql-monitor-agent status**.

> **Warning**
>
> To report its findings, the agent connects 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 5.3.10, “Troubleshooting the Agent”.

5.3.6. 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 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 attempts 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 5.3.1, “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 lets you enable query analysis 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 5.3.1, “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-proxy --plugins-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 contains 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 holds 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
```

5.3.7. 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 5.3, “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 5.3.1, “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 automated discovery, 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 in order for discovery to work properly. For more information, see Section 2.7, “The **Replication** Tab”.

5.3.8. 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 C.2.1, “MySQL Enterprise Agent Configurable Options”.) 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 5.3.5.1, “Starting/Stopping the Agent on Windows”.

- Under Unix see, Section 5.3.5.3, “Starting/Stopping the Agent on Unix”.
- Under Mac OS X see, Section 5.3.5.2, “Starting/Stopping the Agent on Mac OS X”.

5.3.9. Generating a new UUID

For MySQL Enterprise Monitor to operate correctly, each agent must have a unique UUID in order 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

Go to the Agent installation directory and enter the following command:

```
shell> bin/mysql-proxy --plugins=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 Agent installation directory and update the UUID by executing:

```
C:\> cd C:\Program Files\MySQL\Enterprise\Agent
C:\> mysql-monitor-agent -uf mysql-monitor-agent.ini
```

This updates the configuration file directly with the new UUID.

5.3.10. 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 5.3.5.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 5.3.5.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 5.3.5.2, “Starting/Stopping the Agent on Mac OS X”

Some additional tips are noted below:

- If incorrect credentials are specified for the agent login to the MySQL server that it is monitoring, then the agent cannot 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, correct them and restart the agent. Remember, changes to the `ini` file only take effect when the agent is restarted.
- The agent cannot start up if incorrect credentials are specified for the service manager login. Using incorrect credentials for logging into the service manager creates an entry in the agent log file. For the location of this log file see Section C.2.3, “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 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.

- To confirm that the agent can log in to the service manager, 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, 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.2.3, “Agent Log and pid Files”.

  If no HTTP authentication dialog box appears, and you cannot connect at all, you might 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 5.3.8, “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.2.3, “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 C.2.1, “MySQL Enterprise Agent Configurable Options”.

- 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 C.2.2, “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 5.3.7, “Configuring an Agent to Monitor a Remote MySQL Server”.

- The MySQL Enterprise Agent and MySQL Enterprise Service Manager use the hostid, stored within the mysql.inventory table on the monitored MySQL Server to determine whether the instance being monitored is a clone. The hostid of the current server is checked against the stored value when the agent starts. If the generated hostid and stored hostid do not match, you get an error similar to the following in the agent log file:

  ```
  %s: [%s] the hostid from mysql.inventory doesn't match our agent's host-id (%s != %s)
  We assume that this is a cloned host and shutdown now.
  Please TRUNCATE TABLE mysql.inventory on this mysql-instance and restart the agent.
  If this is a master for replication, please also run SET SQL_LOG_BIN = 0; first.
  ```

  To fix the problem, connect to the MySQL server using the credentials configured when you installed the agent, then truncate the mysql.inventory table:

```
mysql> TRUNCATE mysql.inventory;
```

Now restart the agent, which recreates the mysql.inventory table with the updated instance UUID and hostid information.

- If the MySQL Enterprise 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.

5.4. Unattended Installation

It is possible to install or upgrade the MySQL Enterprise Service Manager and MySQL Enterprise Agent components without any direct user interaction. This can be useful if you want to perform an automated installation, or if you need to perform an installation across multiple machines.

To perform an unattended installation, you need to specify the installation mode as unattended by using the mode command line option. Using this mode, you specify the installation parameters, such as the installation directory, and user, password, and network options, by using additional command-line options. Alternatively, 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 performing an unattended installation, it is recommended that you install 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.

5.4.1. Performing an Unattended Installation

The basic process for performing an unattended installation is the same for both the MySQL Enterprise Agent and MySQL Enterprise Service Manager installers, with the only difference being the options supported by each installer. For information on the options for MySQL Enterprise Service Manager, see Section 5.4.2, “MySQL Enterprise Service Manager Options”; for information on the options for MySQL Enterprise Agent, see Section 5.4.3, “MySQL Enterprise Agent Options”.

There are two methods for installation, you can either specify the option on the command line, or use an options file that details the options and their values that you want to be used during installation.

For example, using the command-line method you could install the MySQL Enterprise Agent using:

```
shell> mysqlmonitoragent-version-linux-glibc2.3-x86-64bit-installer.bin
   --installdir /data0/mysql/agent
   --mysqlhost 127.0.0.1 --mysqlport 3306 --mysqluser agent --mysqlpassword mysql --checkmysqlhost yes
   --managerhost localhost --managerport 48080 --agentuser AGENTUSER --agentpassword PASSWORD
   --proxyport 44044 --mode unattended --enableproxy 1
   --createaccount 1 --rootuser ROOTUSER --rootpassword PASSWORD
```

For unattended installation using an option file, you must create a text file that contains the definition for the installation. For this example, the file will be named options.server.txt. An example of the options configuration file is shown below:

```
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.

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, you can start the unattended installation from the command line.

On Windows within a command shell:

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

On Unix you would use a command-line of the form:

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

On Mac OS X, you must locate the `installerbuilder.sh` within the installation package directory. For example:

```
shell> ./mysqlmonitoragent-version-osx-installer.app/Contents/MacOS/installbuilder.sh --optionfile options.server.txt
```

When installing MySQL Enterprise Agent, the same basic process can be followed using the MySQL Enterprise Agent installer and the corresponding agent options.

As a minimum for the MySQL Enterprise 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.

5.4.2. MySQL Enterprise Service Manager Options

The options for the installation for MySQL Enterprise Service Manager are detailed below. The MySQL Enterprise Service Manager supports using a bundled MySQL server, or a separate MySQL server provided by the user. If you are using your own MySQL server, the server must be installed and running before installation. For more information, see Section 5.2.1, “Service Manager Installation Common Parameters”.

Table 5.1. MySQL Enterprise Service Manager Installer Options

Format	Option File	Description	Introduction	Deprecated	Removed
--adminpassword		Password for the database repository			
--adminuser		Username for the database repository			
--dbhost		Hostname or IP address of the MySQL server			
--dbport		TCP/IP port for the MySQL server			
--debuglevel		Set the debug information level			
--debugtrace		File for a debug trace of the installation			
--forceRestart		Forces the server to be restarted	2.2.3		
--help		Display the list of valid options			
--installdir		Installation directory			
--installer-language		Language selection			
--mode		Installation mode			
--mysql_installation_type		MySQL server to be used by the MySQL Enterprise Monitor			
--optionfile		Installation option file			
--tomcatport		Server port for the Tomcat component			
--tomcatshutdown-		Shutdown TCP/IP port for the Tomcat component			

Format	Option File	Description	Introduction	Deprecated	Removed
port					
--tomcatsslport		SSL TCP/IP port for the Tomcat component			
--unattendedmodeui		Unattended mode user interface			
--usessl		Enable SSL support			
--version		Display the product information			

- `--help`

Command-Line Format	`--help`

Display the list of valid installer options.

- `--version`

Display product and version information.

- `--optionfile`

Command-Line Format	`--optionfile`

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

- `--mode`

The installation mode to be used for this installation.

- `--debugtrace`

Command-Line Format	`--debugtrace`	
	Permitted Values	
	Type	`string`

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

- `--debuglevel`

Command-Line Format	`--debuglevel`	
	Permitted Values	
	Type	`numeric`
	Default	`2`
	Range	`0-4`

Set the debug information level for log data written to the file specified by `debugtrace`.

- `--installer-language`

Command-Line Format	`--installer-language`

	Permitted Values		
	Type	string	
	Default	en	
	Valid Values	en	English
		ja	Japanese

The installer language.

- `--installdir`

  The installation directory for MySQL Enterprise Service Manager.

- `--tomcatport`

Command-Line Format	--tomcatport	
	Permitted Values	
	Type	numeric
	Default	18080

The TCP/IP port for the MySQL Enterprise Service Manager. This port will be used by MySQL Enterprise Agent and as the port for the interface to the MySQL Enterprise Dashboard

- `--tomcatshutdownport`

Command-Line Format	--tomcatshutdownport	
	Permitted Values	
	Type	numeric
	Default	18005

The MySQL Enterprise Service Manager Tomcat shutdown port.

- `--tomcatsslport`

Command-Line Format	--tomcatsslport	
	Permitted Values	
	Type	numeric
	Default	18443

The TCP/IP port to be used for SSL communication to the MySQL Enterprise Service Manager

- `--usessl`

Command-Line Format	--usessl	
	Permitted Values	
	Type	boolean
	Default	0

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

- `--adminuser`

Command-Line Format	`--adminuser`		
	Permitted Values		
	Type	`string`	
	Default	`service_manager`	

The user name to be used for connecting to the database repository used by MySQL Enterprise Service Manager. If you are installing the bundled MySQL server, this is the username that will be configured. If you are using an existing MySQL server, this should be the username of the user with rights to access the database.

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.1.5, “The `config.properties` File”.

- `--unattendedmodeui`

Command-Line Format	`--unattendedmodeui`		
	Permitted Values		
	Type	`string`	
	Default	`none`	
	Valid Values	`none`	No dialogs
		`minimal`	Critical dialogs
		`minimalWithDialogs`	Minimal UI with dialogs

The UI elements to use when performing an unattended installation. The options are `none`, show no UI elements during the installation; `minimal`, show minimal elements during installation; `minimalWithDialogs`, show minimal UI elements, but include the filled-dialog boxes.

- `--adminpassword`

Command-Line Format	`--adminpassword`	
	Permitted Values	
	Type	`string`

The MySQL Enterprise Service Manager password for connecting to the MySQL database repository.

- `--mysql_installation_type`

Command-Line Format	`--mysql_installation_type`		
	Permitted Values		
	Type	`string`	
	Default	`bundled`	
	Valid Values	`bundled`	Use the bundled MySQL server
		`existing`	Use an existing (user supplied) MySQL server

Specifies whether the installer should configure MySQL Enterprise Service Manager to install the bundled MySQL server, or use a

MySQL server that you have already installed to store the repository data.

- `--dbport`

Command-Line Format	`--dbport`	
	Permitted Values	
	Type	`numeric`
	Default	`13306`

The TCP/IP port for the MySQL database used to store MySQL Enterprise Service Manager repository data. If you are installing the bundled MySQL server, this is the port that the MySQL server will be configured to use to listen for connections. If you are using an existing MySQL server, this should be the port for the MySQL server that you want to connect to.

- `--dbhost`

Command-Line Format	`--dbhost`	
	Permitted Values	
	Type	`string`
	Default	`127.0.0.1`

The hostname for the MySQL database. When installing MySQL Enterprise Service Manager to use an existing MySQL server, this should be the hostname of the server that will store the database repository.

- `--forceRestart`

Version Introduced	2.2.3	
Command-Line Format	`--forceRestart`	
	Permitted Values	
	Type	`boolean`
	Default	`0`

Force a restart of MySQL Enterprise Service Manager, even if there is a change to the configuration file that would normally prevent the automatic restart of the server.

5.4.3. MySQL Enterprise Agent Options

To view all the options available for an unattended *agent* installation, invoke the agent installer file passing in the `help` option. The available options are detailed in the table below.

Table 5.2. MySQL Enterprise Agent Installer Options

Format	Option File	Description	Introduction	Deprecated	Removed
--agentpassword		Password of the agent user for connecting to the monitored MySQL server			
--agentuser		Username of the agent for connecting to the monitored MySQL server			
--checkmysqlhost		Validate the supplied MySQL hostname			
--createaccount		Create an account on the monitored MySQL server to be used by the agent			
--debuglevel		Set the debug information level			

Format	Option File	Description	Introduction	Deprecated	Removed
--debugtrace		File for a debug trace of the installation			
--enableagent		Enable the agent to monitor a MySQL server			
--enableproxy		Enable query analyzer data to be captured using the proxy service			
--help		Display the list of valid options			
--installdir		Installation directory			
--installer-language		Language selection			
--managerhost		Hostname of IP address of the MySQL Enterprise Monitor server			
--managerport		TCP/IP port of the MySQL Enterprise Monitor server			
--managersslport		TCP/IP port of the MySQL Enterprise Monitor server for SSL communication			
--mode		Installation mode			
--mysqlconnmethod		Connection method to the monitored MySQL server			
--mysqlhost		MySQL hostname/IP address			
--mysqlpassword		MySQL password for the monitored MySQL server			
--mysqlport		TCP/IP port for the monitored MySQL server			
--mysqlsocket		Unix socket/Named pipe for the monitored MySQL server			
--mysqluser		MySQL username for the monitored MySQL server			
--optionfile		Installation option file			
--proxyport		TCP/IP port for the Proxy module			
--proxyuser		User account for the proxy service			
--rootpassword		Password for root user			
--rootuser		User with root privileges on the monitored MySQL server			
--servicename		Service name			
--unattendedmodeui		Unattended mode user interface			
--usessl		Enable use of SSL communication with MySQL Enterprise Monitor server			
--version		Display the product information			

- `--agentpassword`

Command-Line Format	`--agentpassword`	
	Permitted Values	
	Type	`string`

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

- `--agentuser`

Command-Line Format	`--agentuser`

	Permitted Values	
	Type	string
	Default	agent

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

- --checkmysqlhost

Command-Line Format	--checkmysqlhost		
	Permitted Values		
	Type	string	
	Default	yes	
	Valid Values	yes	Check host
		no	Do not check host

Validate the MySQL hostname or IP address

- --createaccount

Command-Line Format	--createaccount	
	Permitted Values	
	Type	boolean
	Default	1

Create the user account specified by the agentuser and agentpassword options to be used when the agent connects to the monitored MySQL server. To use this option you must also specify a suitable root account with privileges to create the new user using the rootuser and rootpassword options.

The user create will have the correct privileges for the agent to collect all the necessary information. For more information, see Section 5.3.1, “Creating a MySQL User Account for the Monitor Agent”.

- --debuglevel

Command-Line Format	--debuglevel	
	Permitted Values	
	Type	numeric
	Default	2
	Range	0-4

Set the debug information level for log data written to the file specified by debugtrace.

- > --debugtrace

Command-Line Format	--debugtrace	
	Permitted Values	
	Type	string

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

- `--enableagent`

Command-Line Format	`--enableagent`	
	Permitted Values	
	Type	`boolean`
	Default	`1`

Enable the agent to monitor a MySQL server. If this option is set to false, the agent will be installed without enabling the monitoring of MySQL server. You can use this to configure a proxy service using an agent for collecting Query Analyzer information.

- `--enableproxy <enableproxy>`

Command-Line Format	`--enableproxy`	
	Permitted Values	
	Type	`boolean`
	Default	`1`

Enable the Proxy service. The proxy service can be used to collect query data to be used through the Query Analyzer. For more information, see Section 3.1.3, “Using the Agent Proxy”.

- `--installdir`

Command-Line Format	`--installdir`	
	Permitted Values	
	Type (windows)	`string`
	Default	`C:\Program Files\MySQL\Enterprise\Agent`
	Permitted Values	
	Type (macosx)	`string`
	Default	`/Applications/mysql/enterprise/agent/`
	Permitted Values	
	Type (unix)	`string`
	Default	`/opt/mysql/enterprise/agent/`

Specify the directory where the software will be installed.

- `--installer-language`

Command-Line Format	`--installer-language`		
	Permitted Values		
	Type	`string`	
	Default	`en`	
	Valid Values	`en`	English
		`ja`	Japanese

Set the language to be used for the installation process.

- `--managerhost`

Command-Line Format	--managerhost	
	Permitted Values	
	Type	string

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

- --managerport

Command-Line Format	--managerport	
	Permitted Values	
	Type	numeric
	Default	18080

The port number of the MySQL Enterprise Service Manager.

- --managersslport

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

- --mode

Command-Line Format	--mode		
	Permitted Values		
	Type (macosx)	string	
	Default	osx	
	Valid Values	osx	Mac OS X (native)
		text	Text (command-line)
		unattended	Unattended (no dialogs/prompts)
	Permitted Values		
	Type (linux)	string	
	Default	gtk	
	Valid Values	gtk	GTK (X Windows)
		xwindow	X Windows (native)
		text	Text (command-line)
		unattended	Unattended (no dialogs/prompts)
	Permitted Values		
	Type (unix)	string	
	Default	xwindow	
	Valid Values	xwindow	X Windows (native)
		text	Text (command-line)
		unattended	Unattended (no dialogs/prompts)

	Permitted Values		
	Type (windows)	`string`	
	Default	`win32`	
	Valid Values	`win32`	Windows (native)
		`unattended`	Unattended (no dialogs/prompts)

Specify the installation mode to use for this installation.

- `--mysqlconnmethod`

Command-Line Format	`--mysqlconnmethod`		
	Permitted Values		
	Type	`string`	
	Default	`tcpip`	
	Valid Values	`tcpip`	Use TCP/IP
		`socket`	Use Unix Socket/Named Pipe

Specify the connection method to use to connect to MySQL. If you specify `tcpip`, the value of the `mysqlport` option will be used. If you specify `socket`, the value of the `mysqlsocket` option will be used to connect to the MySQL server to be monitored.

- `--mysqlhost`

Command-Line Format	`--mysqlhost`	
	Permitted Values	
	Type	
	Default	`127.0.0.1`

Hostname or IP address of the MySQL server to be monitored.

- `--mysqlpassword`

Command-Line Format	`--mysqlpassword`	
	Permitted Values	
	Type	`string`

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

- `--mysqlport`

Command-Line Format	`--mysqlport`	
	Permitted Values	
	Type	`numeric`
	Default	`3306`

The TCP/IP port to use when connecting to the monitored MySQL server.

- `--mysqlsocket`

Command-Line Format	`--mysqlsocket`	
	Permitted Values	
	Type	`string`

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

- `--mysqluser`

Command-Line Format	`--mysqluser`	
	Permitted Values	
	Type	`string`

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

- `--optionfile <optionfile>`

Command-Line Format	`--optionfile`

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

- `--proxyport`

Command-Line Format	`--proxyport`	
	Permitted Values	
	Type	`numeric`
	Default	`6446`

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

- `--proxyuser`

Command-Line Format	`--proxyuser`	
	Permitted Values	
	Type	`string`
	Default	`root`

The user account for the management component of the proxy service.

- `--unattendedmodeui`

Command-Line Format	`--unattendedmodeui`

	Permitted Values		
	Type	`string`	
	Default	`none`	
	Valid Values	`none`	No dialogs
		`minimal`	Critical dialogs
		`minimalWithDialogs`	Minimal UI with dialogs

The UI elements to use when performing an unattended installation. The options are `none`, show no UI elements during the installation; `minimal`, show minimal elements during installation; `minimalWithDialogs`, show minimal UI elements, but include the filled-dialog boxes.

- `--usessl`

Command-Line Format	`--usessl`	
	Permitted Values	
	Type	`boolean`
	Default	`0`

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

- `--version`

Command-Line Format	`--version`

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

- `--rootpassword`

Command-Line Format	`--rootpassword`	
	Permitted Values	
	Type	`string`

The password of a root account to be used to create the agent user by the installer.

- `--rootuser`

Command-Line Format	`--rootuser`	
	Permitted Values	
	Type	`string`
	Default	`root`

The root account to be used to create the agent user by the installer.

- `--servicename`

Command-Line Format	`--servicename`

	Permitted Values	
	Type	`string`
	Default	`mysql.monitor.agent`

When the MySQL Enterprise Agent is installed, a new service is created (Windows), or on Unix a new startup script is created within the corresponding startup directory (for example `/etc/init.d`). If you want to install multiple agents on the same host you can use this option to create each agent installation with a unique identifier. During an upgrade installation, you then use this identifier to specify which installation of the agent should be updated.

- `--help`

Command-Line Format	`--help`

Display the list of valid options to the installer.

5.5. 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 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 C.2, “MySQL Enterprise Agent Reference”.
- **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 5.6.2.1, “Upgrading the Monitored MySQL Server” for instructions on upgrading a server.
- **Repository access**: You may want to add other users.

5.6. 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 5.6.1, “Upgrading MySQL Enterprise Monitor”.
- For more information on re-installing an existing installation, see Reinstalling MySQL Enterprise Monitor.
- To change an existing installation, such as changing the monitored server, see Section 5.6.2, “Changing Your MySQL Enterprise Monitor Installation”.

5.6.1. Upgrading MySQL Enterprise Monitor

From time to time there may be updates to the MySQL Enterprise Service Manager or the MySQL Enterprise 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.

You also cannot use the upgrader if you have configured the MySQL server for the MySQL Enterprise Service Manager instance to be located away from the rest of the MySQL Enterprise Service Manager and installation directory.

The installation and configuration of MySQL Enterprise 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-2.2.0-windows-update-installer.exe`* would indicate a Windows update to MySQL Enterprise Service Manager version 2.2.0.

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 Agent. Once you have shutdown each component, start by updating the MySQL Enterprise Service Manager, and then updating the MySQL Enterprise 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 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 overwrites `items-mysql-monitor.xml`. On Windows, this file is 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. 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 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 5.2.7.3, “Installing, Updating or Upgrading Advisors After Initial Login”.

5.6.1.1. 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 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 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 5.6.1.1.1, “Upgrading to MySQL Enterprise Service Manager 2.0”. For more information on upgrading an MySQL Enterprise Agent, see Section 5.6.1.1.2, “Upgrading to MySQL Enterprise Agent 2.0”.

5.6.1.1.1. 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.

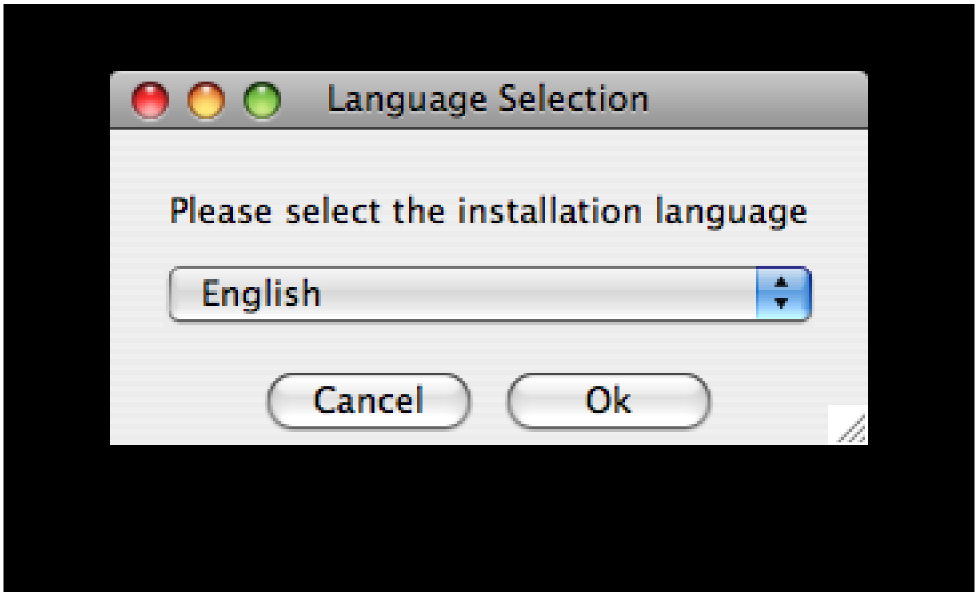
The installation of the new software using the update installer follows this basic sequence:

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 5.20. MySQL Enterprise Monitor: Server Update: Language Selection



Please select the installation language

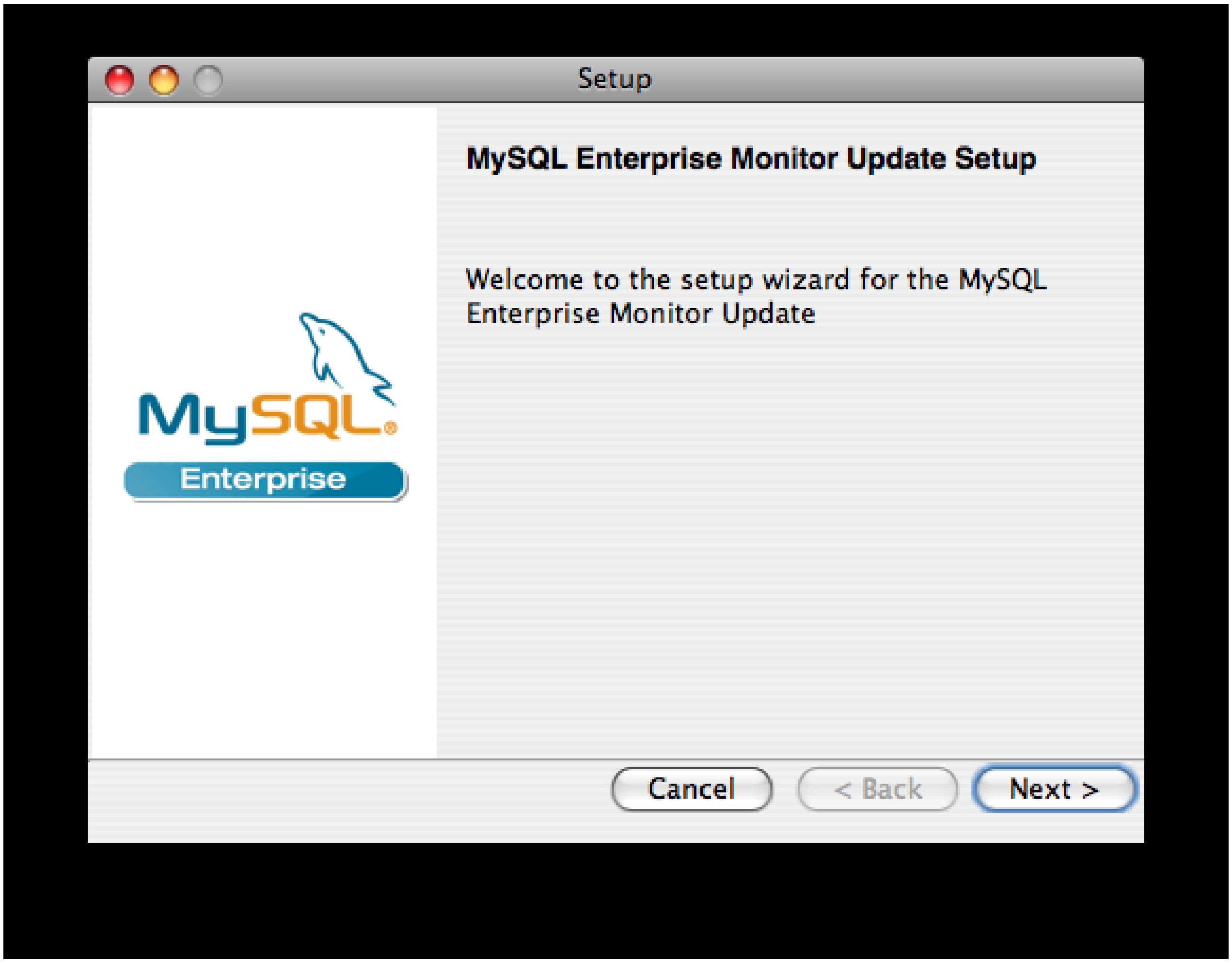
English

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 5.21. MySQL Enterprise Monitor: Server Update: Previous Installation

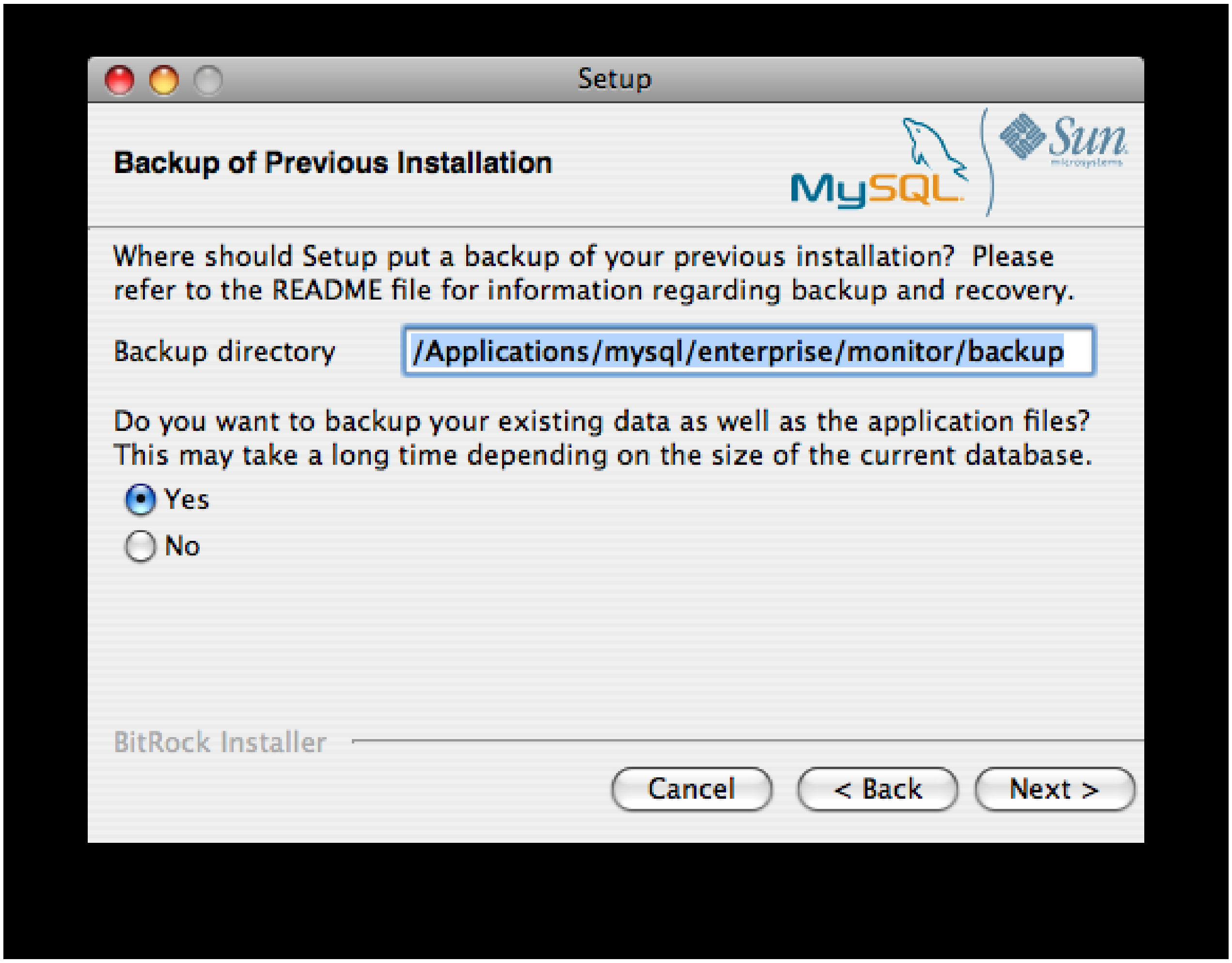


MySQL Enterprise Monitor Update Setup

Welcome to the setup wizard for the MySQL Enterprise Monitor Update

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 5.22. MySQL Enterprise Monitor: Server Update: Backup of Previous Installation



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

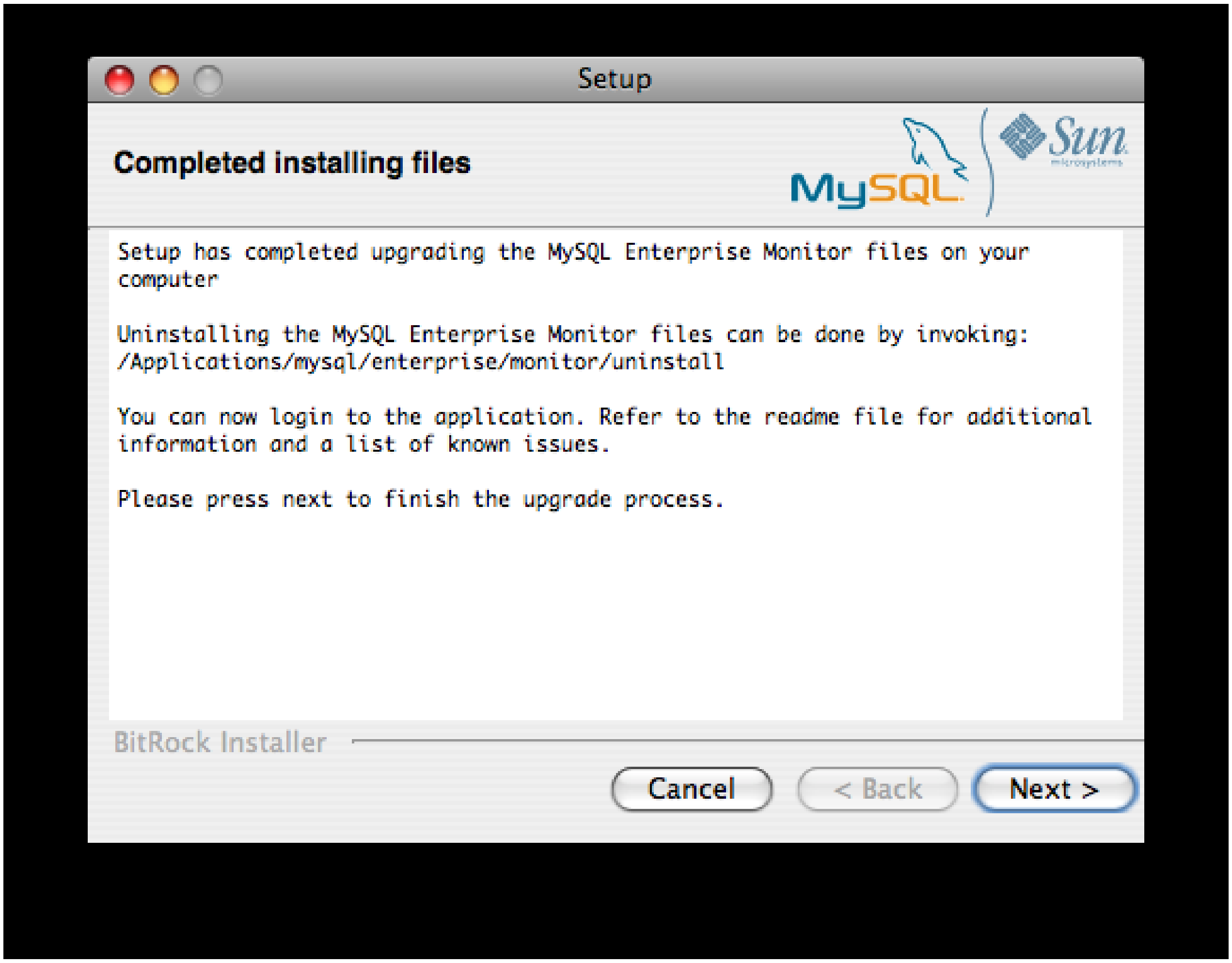
Specify the location of the backup (default is to use the `backup` directory within your installation directory). Note that backing up the database in addition to the main application will increase the installation time as the files have to be copied. The larger the size of your repository data, the 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 5.23. 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.

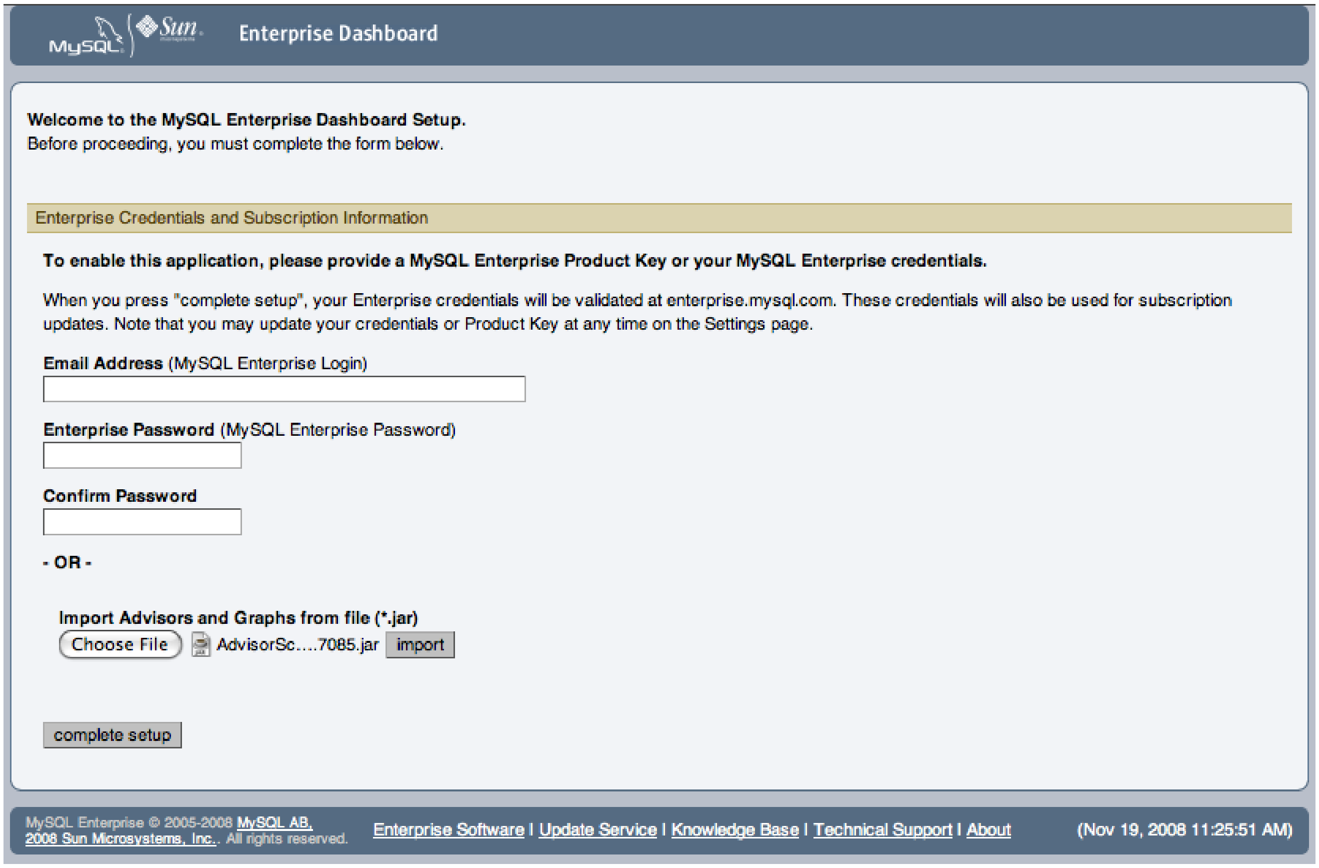
Please press next to finish the upgrade process.

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 5.24. MySQL Enterprise Monitor: Server Update: Final Setup



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.

Enterprise Software | Update Service | Knowledge Base | Technical Support | About

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

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

5.6.1.1.2. Upgrading to MySQL Enterprise 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 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
   ```

   For MySQL Enterprise Agent 2.2 and higher, If you have multiple instances of MySQL Enterprise Agent installed, you can specify a unique name for the agent, which will be used to identify the agent startup script (in `/etc/init.d` and other configuration information. To specify the installation, use the `--servicename=`*`mysql-monitor-agent`* option.

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 lets you monitor the execution statistics 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 Section 2.6, “The **Query Analyzer** Tab”.

```
----------------------------------------------------------------------------
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 [6446]:

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 Agent to your MySQL server you must update the grants for the MySQL Enterprise 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 B.6, “Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0”.

5.6.1.2. Upgrading to MySQL Enterprise Monitor 2.1

When upgrading to MySQL Enterprise Monitor 2.1 you should be aware of the following issues:

- During the upgrade process, if you had previously set a Query Analyzer data purge value that was greater than the standard data purge value, the historical data collection purge period will be set to the Query Analyzer purge period. This will be logged in the upgrade log with the following notice:

  ```
  getDataCollectionPurgeLifespan now: 4 weeks (was: 2 weeks)
  ```

5.6.1.3. Upgrading to MySQL Enterprise Monitor 2.2

When upgrading to MySQL Enterprise Monitor 2.2 you should be aware of the following issues:

- If you are upgrading to MySQL Enterprise Monitor 2.2 you should upgrade from MySQL Enterprise Monitor 2.0 or higher. If you are using MySQL Enterprise Monitor 1.3 or lower, you should upgrade to MySQL Enterprise Monitor 2.0 first before upgrading to MySQL Enterprise Monitor 2.2.

- Rules and advisors in MySQL Enterprise Monitor 2.2 now support auto-close functionality if the issue that raised the notification is no longer causing a problem. You should review all your existing rules schedules and confirm whether the auto-close feature is useful in each case.

- The MySQL Enterprise Service Manager supports the use of a separate, self-managed, MySQL server in place of the MySQL server bundled with the MySQL Enterprise Service Manager installation for storing repository data. For more information, see Section 5.2.1, “Service Manager Installation Common Parameters”.

- If you are upgrading MySQL Enterprise Service Manager which uses a remote, self-managed MySQL database to store the MySQL Enterprise Monitor repository data, then you should perform the installation as follows:

  1. Shutdown MySQL Enterprise Service Manager.

  2. Shutdown the MySQL instance supporting your MySQL Enterprise Monitor repository.

  3. Upgrade the MySQL server as you would any other MySQL server instance. See Upgrading MySQL.

     Ensure when performing the upgrade that the MySQL server instance is configured with support for InooDB and that the InooDB storage engine has been disabled. Also ensure that support for partitioning is available within the server.

  4. Run the `mysql_upgrade` to ensure that the system tables have been correctly updated.

  5. Restart the MySQL repository instance.

  6. Restart MySQL Enterprise Service Manager and confirm that it was able to connect to the MySQL server.

- When upgrading MySQL Enterprise Service Manager, the installer tries to determine if you are using a local MySQL server (as bundled with the MySQL Enterprise Service Manager) or a remote and self-managed MySQL server instance to store your repository data.

To determine whether the existing installation is using a local MySQL server, the installer examines the existing MySQL Enterprise Service Manager configuration file. If the value of the `mysql.server` configuration parameter is set to `127.0.0.1` or `localhost`, the installer assumes the MySQL server is local. If any value is identified, then it is assumed that the server is remote/ self-managed and the installer will not upgrade the MySQL server component.

If the correct MySQL server version and supported functionality (partitioning) are not supported by the MySQL server you are using, the installer will terminate.

- The storage tables used for storing MySQL Enterprise Service Manager data, and the rules and methods required for purging old information, have changed in MySQL Enterprise Service Manager 2.2. You can improve the purging and disk space usage for these data tables by using the techniques shown in Section B.7, "Improving Disk Space Usage with Purge in 2.1 and 2.2".
- For MySQL Enterprise Agent installations, the installer can now create a user on your MySQL server with suitable credentials. To perform this operation, the installer will request the root login and password for your MySQL server so that the user and other details can be updated.

5.6.2. Changing Your MySQL Enterprise Monitor Installation

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

5.6.2.1. Upgrading the Monitored MySQL Server

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

- Windows: Section 5.3.5.1, "Starting/Stopping the Agent on Windows"
- Unix: Section 5.3.5.3, "Starting/Stopping the Agent on Unix"
- Mac OS X: Section 5.3.5.2, "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 5.2.5, "Starting/Stopping the MySQL Enterprise Monitor Service on Windows".

To stop and restart the MySQL daemon under Unix and Mac OS X, see, Section 5.2.6, "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.

5.6.2.2. Changing the Server That an Agent Monitors

You need not reinstall the MySQL Enterprise Agent in order 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 5.3.5.1, "Starting/Stopping the Agent on Windows"
- Unix: Section 5.3.5.3, "Starting/Stopping the Agent on Unix"
- Mac OS X: Section 5.3.5.2, "Starting/Stopping the Agent on Mac OS X"

For instructions on removing a server see, Section 2.8.3.3, "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-monit-`

or-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 C.2.2, "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 C.2.1, "MySQL Enterprise Agent Configurable Options".

To restart the agent see:

- Windows: Section 5.3.5.1, "Starting/Stopping the Agent on Windows"
- Unix: Section 5.3.5.3, "Starting/Stopping the Agent on Unix"
- Mac OS X: Section 5.3.5.2, "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 5.3.7, "Configuring an Agent to Monitor a Remote MySQL Server".
>
> If you experience difficulties starting the agent, check Section 5.3.10, "Troubleshooting the Agent".

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

5.6.2.3. 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 in order 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 5.3.5.1, "Starting/Stopping the Agent on Windows"
- Unix: see Section 5.3.5.3, "Starting/Stopping the Agent on Unix"
- Mac OS X: see Section 5.3.5.2, "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 2.3.3, “Editing Built-in Rules”. To blackout all events associated with a specific server or group of servers see Section 4.4, “Advisor Blackout Periods”.

5.7. Uninstalling the MySQL Enterprise Monitor

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

5.7.1. 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 Monitor` 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 5.2.5, “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 5.2.5, “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.

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 C.2.2, “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 5.3.5.1, “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 C.2.1, “MySQL Enterprise Agent Configurable Options”.

Go to the command line and remove the MySQL Enterprise 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.

5.7.2. 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 5.2.6, “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, including 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 5.3.5.3, “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 C.2.1, “MySQL Enterprise Agent Configurable Options”.

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.

5.7.3. 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 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 Agent you may remove the `/Applications/mysql/enterprise/agent` directory.

Removing the MySQL Enterprise 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 5.3.5.2, “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 C.2.1, “MySQL Enterprise Agent Configurable Options”.

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.

Appendix A. Licenses for Third-Party Components

MySQL Enterprise Monitor 2.2

A.1. Apache Commons BeanUtils

The following software may be included in this product:

```
Apache Commons BeanUtils

Component's NOTICE.txt file:
Apache Commons BeanUtils
Copyright 2000-2008 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.17, “Apache License Version 2.0, January 2004”.

A.2. Apache Commons Codec

The following software may be included in this product:

```
The following software may be included in this product:
Apache Commons Codec
Copyright 2002-2009 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

------
src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data
from http://aspell.net/test/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org).

Verbatim copying and distribution of this entire article is permitted in
any medium, provided this notice is preserved.
------
```

This component is licensed under Section A.17, “Apache License Version 2.0, January 2004”.

A.3. Apache Commons Collections

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.17, “Apache License Version 2.0, January 2004”.

A.4. Apache Commons DBCP

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.17, “Apache License Version 2.0, January 2004”.

A.5. Apache Commons Digester

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.17, “Apache License Version 2.0, January 2004”.

A.6. Apache Commons Exec

The following software may be included in this product:

```
Apache Commons Exec

Component's NOTICE.txt file:
Apache Commons Exec
Copyright 2005-2009 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.17, “Apache License Version 2.0, January 2004”.

A.7. Apache Commons FileUpload

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.17, “Apache License Version 2.0, January 2004”.

A.8. Apache Commons HttpClient

The following software may be included in this product:

```
Apache Commons HttpClient

Component's NOTICE.txt file:
Apache Jakarta HttpClient
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.17, “Apache License Version 2.0, January 2004”.

A.9. Apache Commons IO

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.17, “Apache License Version 2.0, January 2004”.

A.10. Apache Commons Lang

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.17, “Apache License Version 2.0, January 2004”.

A.11. Apache Commons Logging

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.17, “Apache License Version 2.0, January 2004”.

A.12. Apache Commons Pool

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.17, “Apache License Version 2.0, January 2004”.

A.13. Apache Commons Validator

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.17, “Apache License Version 2.0, January 2004”.

A.14. Apache HttpComponents HttpClient

The following software may be included in this product:

```
Apache HttpComponents HttpClient

Component's NOTICE.txt file:
```

```
Apache HttpComponents Client
Copyright 1999-2009 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls. See
http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
```

This component is licensed under Section A.17, “Apache License Version 2.0, January 2004”.

A.15. Apache HttpComponents HttpCore

The following software may be included in this product:

```
Apache HttpComponents HttpCore

Component's NOTICE.txt file:
Apache HttpComponents Core - HttpCore
Copyright 2006-2009 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.17, “Apache License Version 2.0, January 2004”.

A.16. Apache JAMES Mime4J

The following software may be included in this product:

```
The following software may be included in this product:
Apache JAMES Mime4J

Component's NOTICE.txt file:
=========================================================================
==      NOTICE file for use with the Apache License, Version 2.0,      ==
=========================================================================

Apache JAMES Mime4j
Copyright 2004-2008 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product may include/use software, JUnit (http://www.junit.org/),
developed by Kent Beck, Erich Gamma, and David Saff
License: Common Public License Version 1.0
(http://www.opensource.org/licenses/cpl.php)
```

This component is licensed under Section A.17, “Apache License Version 2.0, January 2004”.

```
THIS PRODUCT ALSO INCLUDES THIRD PARTY SOFTWARE REDISTRIBUTED UNDER
THE FOLLOWING LICENSES:
JUnit, Common Public License Version 1.0  (junit-3.8.1.jar)
http://junit.org

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON
PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF
THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:
a) in the case of the initial Contributor, the initial code and
documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:
  i) changes to the Program, and

  ii) additions to the Program; where such changes and/or additions to
the Program originate from and are distributed by that particular
Contributor. A Contribution 'originates' from a Contributor if it was
added to the Program by such Contributor itself or anyone acting on
such Contributor's behalf. Contributions do not include additions to
the Program which:

(i) are separate modules of software distributed in conjunction with
```

the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:
a) it must be made available under this Agreement; and
b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED 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. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), 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 OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

```
Everyone is permitted to copy and distribute copies of this Agreement,
but in order to avoid inconsistency the Agreement is copyrighted and
may only be modified in the following manner. The Agreement Steward
reserves the right to publish new versions (including revisions) of
this Agreement from time to time. No one other than the Agreement
Steward has the right to modify this Agreement. IBM is the initial
Agreement Steward. IBM may assign the responsibility to serve as the
Agreement Steward to a suitable separate entity. Each new version of
the Agreement will be given a distinguishing version number. The
Program (including Contributions) may always be distributed subject to
the version of the Agreement under which it was received. In addition,
after a new version of the Agreement is published, Contributor may
elect to distribute the Program (including its Contributions) under
the new version. Except as expressly stated in Sections 2(a) and 2(b)
above, Recipient receives no rights or licenses to the intellectual
property of any Contributor under this Agreement, whether expressly,
by implication, estoppel or otherwise. All rights in the Program not
expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and
the intellectual property laws of the United States of America. No
party to this Agreement will bring a legal action under this Agreement
more than one year after the cause of action arose. Each party waives
its rights to a jury trial in any resulting litigation.

Bnd, Bundle Tool http://www.aqute.biz/Code/Bnd, The Apache License, Version 2.0
ALL OTHERS JARS, BY APACHE SOFTWARE FOUNDATION
ALL OF THESE ARE LICENSED UNDER The Apache License, Version 2.0 EXCEPT:

Apache Commons Logging,
The Apache Software License, Version 1.1 (commons-logging-1.1.1.jar)

The Apache Software License, Version 1.1

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 end-user documentation included with the redistribution, if
any, must include the following acknowledgment: "This product includes
software developed by the Apache Software Foundation
(http://www.apache.org/)."  Alternately, this acknowledgment may
appear in the software itself, if and wherever such third-party
acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" must not be
used to endorse or promote products derived from this software without
prior written permission. For written permission, please contact
apache@apache.org.

5. Products derived from this software may not be called "Apache", nor
may "Apache" appear in their name, without prior written permission of
the Apache Software Foundation.

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 APACHE SOFTWARE FOUNDATION 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.
```

A.17. 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.18. Apache log4j

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.17, “Apache License Version 2.0, January 2004”.

A.19. Apache Tomcat

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.17, “Apache License Version 2.0, January 2004”.

A.20. ASM

The following software may be included in this product:

```
ASM

Copyright (c) 2000-2005 INRIA, France Telecom
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.21. 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.22. JDOM Project

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.23. jQuery

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.24. jQuery UI

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.25. `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.26. `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.27. `LuaFileSystem` Library License

The following software may be included in this product:

```
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.28. OpenSSL v0.9.8o and later

The following software may be included in this product:

```
OpenSSL v0.9.8o and later

Copyright (c) 1998-2008 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.29. Spring Framework

The following software may be included in this product:

```
Spring Framework

Copyright (c) 2004-2009 Rod Johnson, Juergen Hoeller, Keith Donald, Colin Sampaleanu, Rob
Harrop, Alef Arendsen, Thomas Risberg, Darren Davison, Dmitriy Kopylenko, Mark Pollack,
Thierry Templier, Erwin Vervaet, Portia Tung, Ben Hale, Adrian Colyer, John Lewis, Costin
Leau, Mark Fisher, Sam Brannen, Ramnivas Laddad, Arjen Poutsma, Chris Beams, Tareq
Abedrabbo, Andy Clement

Component's NOTICE.txt file:
  ======================================================================
  == NOTICE file corresponding to section 4 d of the Apache License,  ==
  == Version 2.0, in this case for the Spring Framework distribution. ==
  ======================================================================

  This product includes software developed by
  the Apache Software Foundation (http://www.apache.org).

  The end-user documentation included with a redistribution, if any,
  must include the following acknowledgement:

    "This product includes software developed by the Spring Framework
     Project (http://www.springframework.org)."

  Alternatively, this acknowledgement may appear in the software itself,
  if and wherever such third-party acknowledgements normally appear.

  The names "Spring" and "Spring Framework" must not be used to
  endorse or promote products derived from this software without
  prior written permission. For written permission, please contact
  rod.johnson@interface21.com or juergen.hoeller@interface21.com.
  ======================================================================
```

This component is licensed under Section A.17, “Apache License Version 2.0, January 2004”.

A.30. `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 General Reference

This appendix contains reference information for MySQL Enterprise Monitor.

B.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 Agent when operating for Query Analyzer as a proxy cannot handle queries greater than 16MB.
- The MySQL Enterprise 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 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 is 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 Agent, if any of the MySQL instances are down at the point when the agent is started, the agent does 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.

B.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
- Google Chrome 4.0

The MySQL Enterprise Dashboard is known *not* to work within the following browser environments:

- Microsoft Internet Explorer 6.x
- Opera
- Google Chrome

B.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 increases with time to record this historical data. You can control how long information is stored once the service manager has been installed.

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)

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.

B.4. 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, shut down 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.

B.5. 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.

B.6. 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 is migrated. Data older than the purge period is 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.

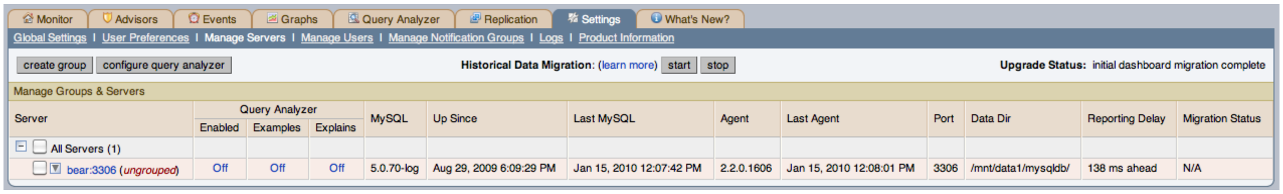
  With the last item, where the migration never completes, occasionally there are some aspects of the data that cannot be migrated successfully. This prevents the migration process completing, but does not affect the conversion of any data that could be migrated.

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** are visible.

Figure B.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)

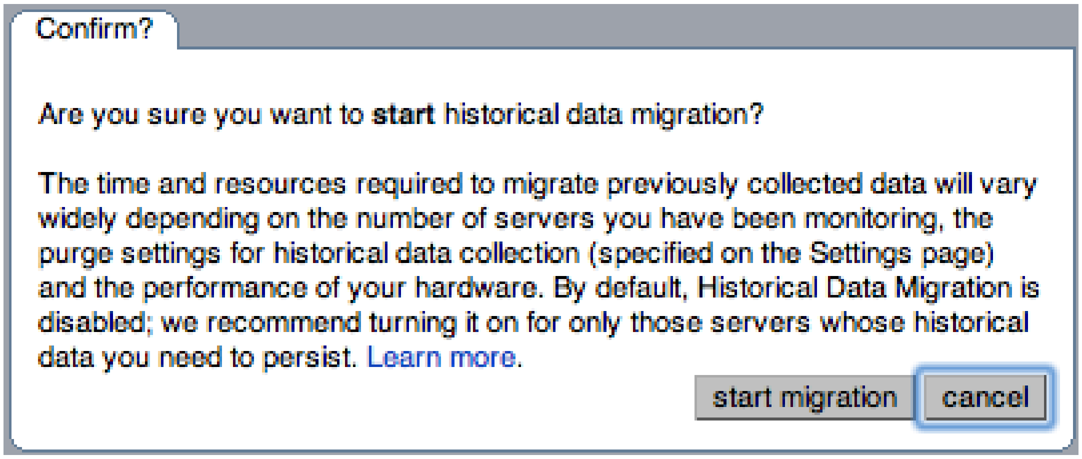
Upgrade Status: initial dashboard migration complete

Manage Groups & Servers

Server	Query Analyzer			MySQL	Up Since	Last MySQL	Agent	Last Agent	Port	Data Dir	Reporting Delay	Migration Status
	Enabled	Examples	Explains									
⊟ □ All Servers (1)												
□ bear:3306 (*ungrouped*)	Off	Off	Off	5.0.70-log	Aug 29, 2009 6:09:29 PM	Jan 15, 2010 12:07:42 PM	2.2.0.1606	Jan 15, 2010 12:08:01 PM	3306	/mnt/data1/mysqldb/	138 ms ahead	N/A

- 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 show their migration status within the **Migration Status** column. If the server cannot be migrated, `N/A` is shown.
- Click Start next to **Historical Data Migration**.
- To start the migration, click Start Migration in the confirmation dialog box. To cancel migration, click Cancel.
- The servers to migrate show `Queued for Migration` in the **Migration Status** column.

Figure B.2. MySQL Enterprise Monitor: Confirming Historical Data Migration



Confirm?

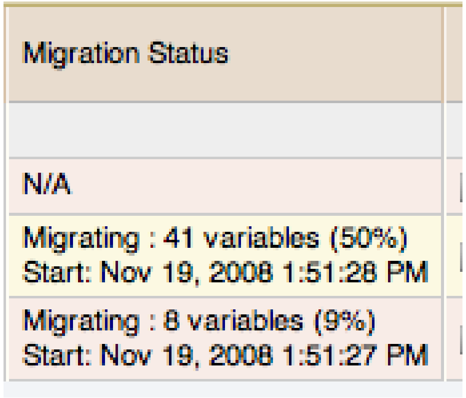
Are you sure you want to **start** historical data migration?

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 B.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**.

You receive confirmation that the migration has been stopped, or a notification if migration has already completed.

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 creates a file, data-1.3.sql containing 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 are removed from the **Manage Servers** panel within MySQL Enterprise Service Manager. To remove access, REVOKE access to the merlin database:

```
mysql& REVOKE ALL on merlin.* FROM 'service_manager';
```

Revoking access to the old data does 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.

B.7. Improving Disk Space Usage with Purge in 2.1 and 2.2

In MySQL Enterprise Monitor 2.1, the data received by the MySQL Enterprise Service Manager from MySQL Enterprise Agent instances is inserted into tables, with the information within the tables purged periodically to remove the old data, according to the defined purging interval.

In installations with a large volume of data, the purging of old information can take a considerable amount of time, slowing down the MySQL Enterprise Service Manager and using up valuable disk space.

In MySQL Enterprise Monitor 2.2, the tables used for storing this information are changed, and the older tables are used only when viewing historical information. The information within the table is not purged until all of the table data is older than the specified purge period, when the old tables are dropped.

> **Note**
>
> The following instructions are designed to reduce the disk space usage for customers that are either using MySQL Enterprise Monitor 2.1 currently, or that have already upgraded an existing MySQL Enterprise Monitor 2.1 installation to MySQL Enterprise Monitor 2.2. If you have performed a new (not upgrade) installation of MySQL Enterprise Monitor 2.2, these steps are not required.

To reduce the space used by the tables in both 2.1 and 2.2 until the data is finally purged, you should follow these steps:

1. If you are running MySQL Enterprise Monitor 2.1, upgrade to MySQL Enterprise Monitor 2.2 using an upgrade installer. This migrates the existing table data and prepare you for the next stage.

   > **Warning**
   >
   > Performing a upgrade installation from MySQL Enterprise Monitor 2.1 to MySQL Enterprise Monitor 2.2 requires approximately 150% of your current disk space requirements while the information is migrated. Once the migration is complete, your disk space usage returns to a value slightly higher than within your MySQL Enterprise Monitor 2.1 installation.

2. Start up your MySQL Enterprise Service Manager and MySQL Enterprise Agent instances.

3. Connect to the MySQL server being used by your MySQL Enterprise Service Manager. You can find a sample `mysql` command for connecting to your server within the `configuration_report.txt` file within your MySQL Enterprise Service Manager installation.

4. Once connected to MySQL, change to the `mem` database:

   ```
   mysql> use mem;
   ```

5. Now execute the following `ALTER TABLE` statements to change the original InnoDB tables to use MyISAM:

   ```
   ALTER TABLE dc_ng_double_age0 ENGINE=MYISAM;
   ALTER TABLE dc_ng_double_age1 ENGINE=MYISAM;
   ALTER TABLE dc_ng_double_age2 ENGINE=MYISAM;
   ALTER TABLE dc_ng_double_now ENGINE=MYISAM;
   ALTER TABLE dc_ng_long_age0 ENGINE=MYISAM;
   ALTER TABLE dc_ng_long_age1 ENGINE=MYISAM;
   ALTER TABLE dc_ng_long_age2 ENGINE=MYISAM;
   ALTER TABLE dc_ng_long_now ENGINE=MYISAM;
   ALTER TABLE dc_ng_string_now ENGINE=MYISAM;
   ```

   > **Note**
   >
   > Please keep in mind the following points when running these statements
   >
   > - Performing this operation should not affect the operation of your MySQL Enterprise Service Manager, since new data is written into different tables.
   > - The time required to perform the conversation depends on the size of your original tables. It may take a considerable amount of time if you have a large volume of old data.
   > - You need slightly more than twice the amount of diskspace currently being used by the tables until the conversion completes.

Once the conversion has completed, you can continue to use your MySQL Enterprise Service Manager without restarting or any further administration. The existing old table data is removed as soon as the data in the old table is older than your selected purging interval. The new MyISAM tables should take up approximately 60% of the space of the InnoDB tables.

B.8. Regular MySQL Enterprise Monitor Maintenance

MySQL Enterprise Monitor is generally self-managing and does not need much maintenance. To keep your MySQL Enterprise Monitor running efficiently, automate or manually perform these maintenance tasks:

- 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.

B.9. Advisor/Graph Reference

Basic

- **Advisors**
  - Agent Host Time Out of Sync Relative to Dashboard
  - 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
- Binary Logs Automatically Removed Too Quickly
- 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
- Event Scheduler Disabled
- 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 Fast Shutdown Enabled
- InnoDB File Format Check Disabled Or Incorrect
- InnoDB INFORMATION_SCHEMA Plugins Missing
- InnoDB Not Using Newest File Format
- InnoDB Strict Mode Is Off
- InnoDB Tablespace Cannot Automatically Expand
- InnoDB Transaction Logs Not Sized Correctly
- 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
- MySQL Server Has Been Restarted
- 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
- Row-based Replication Broken For UTF8 CHAR Columns Longer Than 85 Characters
- Security Alterations Detected: User Privileges Granted
- Security Alterations Detected: User Privileges Revoked
- Security Alterations Have Been Detected
- Security Risk with BINLOG Statement
- 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
- UDF Support Can Be Used To Execute Arbitrary Code
- UDFs Loaded From Insecure Location
- 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 Has Rights To Table 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
  - Warnings Not Being Logged
  - XA Distributed Transaction Support Enabled For InnoDB
- **Graphs**
  - Connections - Aborted
  - Connections - Maximum
  - InnoDB Row Details
  - KBytes In/Out
  - Load Average
  - Memory Usage - Agent
  - Memory Usage - OS Resident
  - Row Access Statistics
  - 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 Error: Unknown or Incorrect Time Zone
  - 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**
  - Connections - Cache
  - InnoDB Buffer Pool
  - InnoDB Compression Time
  - InnoDB Transaction History
  - InnoDB Transactions
  - Memory Usage - OS Virtual
  - MyISAM Key Buffer Usage
  - Opened Tables
  - Query Cache Blocks
  - Query Cache Memory
  - Replication Delay
  - Temporary Tables

Platinum

- **Advisors**
  - AUTO_INCREMENT Field Limit Nearly Reached
  - Binary Log Usage Exceeding Disk Cache Memory Limits
  - Data Flushed To Disk After Each SQL Statement
  - Excessive Disk Temporary Table Usage Detected
  - Excessive Number of Locked Processes
  - Excessive Number of Long Running Processes

- Excessive Number of Long Running Processes Locked
- 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
- MyISAM Indexes Found with No Statistics
- 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
- Tables Found with No Primary or Unique Keys
- Thread Cache Not Enabled
- Too Many Concurrent Queries Running

- **Graphs**
  - InnoDB Adaptive Hash Index Memory
  - InnoDB Adaptive Hash Index Searches
  - InnoDB OS File Access
  - InnoDB Semaphores
  - Query Cache Efficiency
  - Query Cache Lowmem Prunes
  - Query Cache Queries
  - Sort Activity
  - Table Lock Wait Ratio
  - Table Locks

B.9.1. 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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Agent Host Time Out of Sync Relative to Dashboard**

  To maintain data consistency and to facilitate day-to-day system management operations, which often require comparing logs, files, and timestamps across servers, it is important that the server clocks across all your systems and data centers be synchronized with respect to each other relative to UTC time (which takes timezones into account). When the clock on one server is minutes or hours behind another server, any timestamps created in the databases or on the file systems of those two servers will differ by that amount. Thus if you depend on timestamps to test the freshness of some data item, or if you are trying to diagnose a problem and need to compare timestamps across systems, your task will be more complicated due to this time difference.

  In addition, the data and graphs shown in the MySQL Enterprise Monitor Dashboard will be skewed by the difference in time between the machine hosting the Service Manager and the machines running the Agents. For example, if the time on an Agent machine is one hour behind the time on the Service Manager machine, the MySQL server that Agent is monitoring will appear to be down (see Bug#45937) and no data will appear in the graphs for that server for the first hour after the Agent is started.

  Default frequency 00:05:00

  Default auto-close enabled yes

- **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

  Default auto-close enabled no

- **AUTO_INCREMENT Field Limit Nearly Reached**

  Many applications need to generate unique numbers and sequences for identification purposes (e.g. customer IDs, bug or trouble ticket tags, membership or order numbers, etc). MySQL's mechanism for doing this is the AUTO_INCREMENT column attribute, which enables you to generate sequential numbers automatically.

  However, the range of numbers that can be generated is limited by the underlying data type. For example, the maximum value possible for a TINYINT UNSIGNED column is 255. If you try to generate a number that exceeds the maximum allowed by the underlying data type (e.g. by inserting a NULL value into the AUTO_INCREMENT column), you will trigger database errors and your application may not behave properly.

Note that the primary purpose of AUTO_INCREMENT in MySQL is to generate a sequence of `positive` integers. The use of non-positive numbers in an AUTO_INCREMENT column is unsupported, so you may as well define those columns to be UNSIGNED, which effectively doubles their allowable range.

Default frequency 06:00:00

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Binary Logs Automatically Removed Too Quickly**

  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 is used on master replication servers as a record of the statements to be sent to slave servers. It also enables you to review all alterations made to your database.

  However, the number of log files and the space they use can grow rapidly, especially on a busy server, so it is important to remove these files on a regular basis when they are no longer needed, as long as appropriate backups have been made. The `expire_logs_days` parameter enables automatic binary log removal.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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 auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Event Scheduler Disabled**

  The Event Scheduler is a very useful feature when enabled. It is a framework for executing SQL commands at specific times or at regular intervals. Conceptually, it is similar to the idea of the Unix crontab (also known as a "cron job") or the Windows Task Scheduler.

  The basics of its architecture are simple. An event is a stored routine with a starting date and time, and a recurring tag. Once defined and activated, it will run when requested. Unlike triggers, events are not linked to specific table operations, but to dates and times. Using the event scheduler, the database administrator can perform recurring events with minimal hassle. Common uses are the cleanup of obsolete data, the creation of summary tables for statistics, and monitoring of server performance and usage.

Default frequency 00:05:00

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Excessive Number of Locked Processes**

  Depending on the circumstances, storage engines, and other factors, one process may be using or accessing a resource (e.g. a table or row) required by another process in such a way that the second process cannot proceed until the first process releases the resource. In this case the second process is in a "locked" state until the resource is released. If many processes are in a locked state it may be a sign of serious trouble related to resource contention, or a long running session that is not releasing currently held locks when it should have.

  Default frequency 00:01:00

  Default auto-close enabled no

- **Excessive Number of Long Running Processes**

  Most applications and databases are designed to execute queries very quickly. If many queries are taking a long time to execute (e.g. more than a few seconds) it can be a sign of trouble. In such cases queries may need to be tuned or rewritten, or indexes added to improve performance. In other cases the database schema may have to be redesigned.

  Default frequency 00:01:00

  Default auto-close enabled no

- **Excessive Number of Long Running Processes Locked**

  Most applications and databases are designed to execute queries very quickly, and to avoid resource contention where one query is waiting for another to release a lock on some shared resource. If many queries are locked and taking a long time to execute (e.g. more than a few seconds), it can be a sign of performance trouble and resource contention. In such cases queries may need to be tuned or rewritten, or indexes added to improve performance. In other cases the database schema may have to be redesigned.

  Default frequency 00:01:00

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **InnoDB Fast Shutdown Enabled**

  Once you use the InnoDB Plugin on a set of database files, care must be taken to avoid crashes and corruptions when using those files with an earlier version of InnoDB, as might happen by opening the database with MySQL when the plugin is not installed. It is `strongly` recommended that you use a "slow shutdown" (SET GLOBAL innodb_fast_shutdown=0) when stopping the MySQL server when the InnoDB Plugin is enabled. This will ensure log files and other system information written by the plugin will not cause problems when using a prior version of InnoDB.

  The reason "slow" shutdown (innodb_fast_shutdown=0) is recommended is because the InnoDB Plugin may write special records to the transaction undo log that will cause problems if the built-in InnoDB in MySQL attempts to read the log. Specifically, these special records will be written when a record in a COMPRESSED or DYNAMIC table is updated or deleted and the record contains columns stored off-page. The built-in InnoDB in MySQL cannot read these undo log records. Furthermore, the built-in InnoDB in MySQL will fail to roll back incomplete transactions that affect tables that it is unable to read (tables in COMPRESSED or DYNAMIC format).

  Also note that a "normal" shutdown will not necessarily empty the undo log. A normal shutdown occurs when innodb_fast_shutdown=1, the default. When InnoDB is shut down, some active transactions may have uncommitted modifications, or they may be holding a read view that prevents the purging of some version information from the undo log. The next time InnoDB is started after a normal shutdown (innodb_fast_shutdown=1), it will roll back any incomplete transactions and purge old version information. Therefore, it is important to perform a "slow" shutdown (innodb_fast_shutdown=0) as part of the downgrade process.

  Default frequency 12:00:00

Default auto-close enabled no

- **InnoDB File Format Check Disabled Or Incorrect**

  To prevent possible crashes or data corruptions when the InnoDB Plugin opens an ib-file set, it will check that it can fully support the file formats in use within the ib-file set. If the system is restarted following a crash, or a "fast shutdown" (i.e., innodb_fast_shutdown is greater than zero), there may be on-disk data structures (such as redo or undo entries, or doublewrite pages) that are in a "too-new" format for the current software. During the recovery process, serious damage can be done to your data files if these data structures are accessed. The startup check of the file format occurs before any recovery process begins, thereby preventing the problems described in the "Possible Problems" section of the InnoDB Plugin documentation.

  Setting innodb_file_format_check to OFF, or to a different format than the one in use, is very dangerous, as it permits the recovery process to run, possibly corrupting your database if the previous shutdown was a crash or "fast shutdown". You should only set innodb_file_format_check in this manner if you are sure that the previous shutdown was done with innodb_fast_shutdown=0, so that essentially no recovery process will occur.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **InnoDB INFORMATION_SCHEMA Plugins Missing**

  Several INFORMATION_SCHEMA tables -- INNODB_CMP, INNODB_CMPMEM, INNODB_TRX, INNODB_LOCKS, and INNODB_LOCK_WAITS -- contain live information about compressed InnoDB tables, the compressed InnoDB buffer pool, all transactions currently executing inside InnoDB, the locks that transactions hold and those that are blocking transactions waiting for access to a resource (a table or row). These tables are very useful for monitoring the activity and performance of the InnoDB Plugin storage engine.

  However, these INFORMATION_SCHEMA tables are themselves plugins to the MySQL server. As such, they need to be INSTALLed as described in the InnoDB Plugin User's Guide. If they are not installed, you will not be able to use them to monitor the InnoDB storage engine.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **InnoDB Not Using Newest File Format**

  The InnoDB Plugin has two new features that can be very useful -- compressed tables and long variable-length columns stored off-page. Under the right circumstances, both of these features can improve the performance of your system. However, in order to take advantage of these new features you must configure InnoDB to use the new file format that enables them.

  Default frequency 12:00:00

  Default auto-close enabled no

- **InnoDB Strict Mode Is Off**

  To guard against ignored typos and syntax errors in SQL, or other unintended consequences of various combinations of operational modes and SQL commands, the InnoDB Plugin provides a "strict mode" of operations. In this mode, InnoDB will raise error conditions in certain cases, rather than issue a warning and process the specified command (perhaps with some unintended defaults). This is analogous to MySQL's sql_mode, which controls what SQL syntax MySQL will accept, and determines whether it will silently ignore errors, or validate input syntax and data values.

  Using the new clauses and settings for ROW_FORMAT and KEY_BLOCK_SIZE on CREATE TABLE and ALTER TABLE commands and the CREATE INDEX command can be confusing when not running in strict mode. Unless you run in strict mode, InnoDB will ignore certain syntax errors and will create the table or index, with only a warning in the message log. However if InnoDB strict mode is on, such errors will generate an immediate error and the table or index will not be created, thus saving time by catching the error at the time the command is issued.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **InnoDB Transaction Logs Not Sized Correctly**

  To avoid frequent checkpoint activity and reduce overall physical I/O, which can slow down write-heavy systems, the InnoDB transaction 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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **MyISAM Indexes Found with No Statistics**

  The MySQL optimizer needs index statistics to help make choices about whether to use indexes to satisfy SQL queries. Having no statistics or outdated statistics limits the optimizer's ability to make smart and informed access plan choices.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled yes

- **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

  Default auto-close enabled yes

- **MySQL Server Has Been Restarted**

  To perform useful work, a database server must be up-and-running continuously. It is normal for a production server to run continuously for weeks, months, or longer. If a server has been restarted recently, it may be the result of planned maintenance, but it may also be due to an unplanned event that should be investigated.

  Default frequency 00:05:00

  Default auto-close enabled no

- **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

  Default auto-close enabled yes

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled yes

- **Row-based Replication Broken For UTF8 CHAR Columns Longer Than 85 Characters**

  Due to Bug#37426, row-based replication breaks when CHAR() UTF8 fields with a length greater than 85 characters are used.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Security Risk with BINLOG Statement**

  Due to Bug#31611, any user can execute BINLOG statements, which effectively gives them the ability to execute any SQL statement regardless of the privileges associated with their user account (i.e. as given by the GRANT statement). This allows any connected user to get any privileges they want, edit any data they want, add and drop tables, etc.

  This bug has been fixed in later versions of the MySQL server.

  Default frequency 06:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled yes

- **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

  Default auto-close enabled yes

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Slave Error: Unknown or Incorrect Time Zone**

  In order to use time zone names in conjunction with certain statements, functions, and data types, you must configure the server to understand those names by loading information from the operating system's time zone files into a set of tables in the mysql database. However, while the MySQL installation procedure creates those time zone tables, it does not load them; they must be loaded manually after installation.

  Default frequency 00:05:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled yes

- **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

  Default auto-close enabled no

- **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. As an alternative, you should set the `read_only` flag `ON` so the server allows no updates except from users that have the SUPER privilege or from updates performed by slave threads.

  Default frequency 06:00:00

  Default auto-close enabled no

- **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 latest data that was written to the master).

  Default frequency 00:05:00

  Default auto-close enabled no

- **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

  Default auto-close enabled yes

- **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

  Default auto-close enabled yes

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled yes

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled yes

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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 simultaneously 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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Tables Found with No Primary or Unique Keys**

  A primary or unique key of a relational table uniquely identifies each record in the table. Except in very unusual circumstances, every database table should have one or more columns designated as the primary key or as a unique key, and it is common practice to declare one.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **UDF Support Can Be Used To Execute Arbitrary Code**

  In versions 4.1 and 5.0 of MySQL, by default, User Defined Functions (UDFs) are loaded from the system library path (e.g. /usr/lib). It has been determined that the security filter used does not guard properly against certain attacks using existing system libraries. As a result, it is possible for users with privileged access to execute arbitrary code. This problem can also be exploited on systems that are not actively using UDFs, if any untrusted remote users have DBA privileges on MySQL. To properly protect against this problem, a new variable has been introduced, `plugin_dir`, which can be used to specify a different directory from which to load plugins.

  Default frequency 12:00:00

  Default auto-close enabled no

- **UDFs Loaded From Insecure Location**

  User Defined Functions (UDFs) allow you to add features and extend the functionality of your MySQL server, but they also pose a danger if they can be loaded from an insecure location.

  To protect against this problem the `plugin_dir` variable was introduced, which can be used to specify the directory from which to load plugins. If the value is non-empty, user-defined function object files must be located in that directory. If the value is empty, the UDF object files can be located in any directory that is searched by your system's dynamic linker, which does not guard properly against certain attacks using existing system libraries. As a result, it is possible for users with privileged access to execute arbitrary code. This problem can also be exploited on systems that are not actively using UDFs, if any untrusted remote users have DBA privileges on MySQL.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

Default auto-close enabled no

- **User Has Rights To Table That Does Not Exist**

  When a table is dropped, user privileges on the table are not automatically dropped. This has security implications as that user will regain privileges if a table with the same name in the same database is created in the future, which may not be the intended result.

  Default frequency 00:05:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **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

  Default auto-close enabled no

- **Warnings Not Being Logged**

  Error conditions encountered by a MySQL server are always logged in the error log, but warning conditions are only logged if `log_warnings` is set to a value greater than 0. If warnings are not logged you will not get valuable information about aborted connections and various other communication errors. This is especially important if you use replication so you get more information about what is happening, such as messages about network failures and reconnections.

  Default frequency 12:00:00

  Default auto-close enabled no

- **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

  Default auto-close enabled no

B.9.2. Graph Definition Reference

Individual graphs are defined below.

Connections

Displays the number of client connections by displaying the number of connected threads, active threads, and cached threads.

Name	Connections
Range Label	# connections
Series	**Expression**
Cached	`cached`
Running	`running`
Total	`connected`
Variables	**Data Item**
connected	`mysql:status:Threads_connected`
running	`mysql:status:Threads_running`
cached	`mysql:status:Threads_cached`

Connections - Aborted

Shows the number of client connections that have been aborted, either during connection, or aborted active connections. This information is useful to determine if your server is overloaded (and therefore aborting connections), or if there is a network fault. Aborted connections also indicate those which did not disconnect properly, for example during a timeout.

Name	Connections - Aborted
Range Label	total/min
Series	**Expression**
Clients	`clients`
Connections	`connections`
Variables	**Data Item**
clients	`mysql:status:Aborted_clients`
connections	`mysql:status:Aborted_connects`

Connections - Cache

Displays the thread cache information, comparing new thread creations against all database connections.

Name	Connections - Cache
Range Label	total/min
Series	**Expression**
Connections	`connections`
Threads Created	`tcreated`
Variables	**Data Item**
tcreated	`mysql:status:Threads_created`
connections	`mysql:status:Connections`

Connections - Maximum

Shows the maximum number of connections, and the maximum number of used connections.

Name	Connections - Maximum
Range Label	total
Series	**Expression**
Allowed	`max_connections_allowed`

Used	max_connections_used
Variables	**Data Item**
max_connections_allowed	mysql:variables:max_connections
max_connections_used	mysql:status:Max_used_connections

CPU Utilization

Displays the CPU usage statistics over time for the machine hosting the MySQL server. The graph shows the aggregate system, user, wait and idle times for your CPU(s). You can use this information to help determine your CPU utilization and whether you are getting the most efficient use of your CPU power.

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

Displays the database activity, by showing the individual counts over time of different DML statements per minute. For example, you can monitor the number of SELECT statements executed per minute over time and gain an understanding of your server throughput. This information can be used to monitor the activity level, and provide a useful quick reference for your server operation.

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**
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
selects	mysql:status:Com_select
inserts	mysql:status:Com_insert

insert_selects	mysql:status:Com_insert_select
updates	mysql:status:Com_update

Hit Ratios

Shows the cache hit ratios (as a percentage) for the query cache, MyISAM and InnoDB tables, and the connection and thread cache information. The information provided by the hit ratios can provide indicators of whether your cache is working effectively, or is of a size suitable for the data that you are storing and retrieving. Low hit ratios indicate that your cache are inefficient and may need to be investigated.

Name	Hit Ratios
Range Label	%
Series	**Expression**
InnoDB Buffer	`100-((iReads / (iReadRequests+1))*100)`
Key Cache	`(100-((keyReads / (keyReadRequests+1))*100)) * (keyReadRequests >= keyReads)`
Query Cache	`(qHits/(qHits+qInserts+1))*100`
Thread Cache	`100-((tcreated / (connections+1))*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 Adaptive Hash Index Memory

Shows the size of the InnoDB adaptive hash index. These are used to hold the data from tables in memory providing the size of the indexes is small enough to fit in memory. The graph shows the memory usage of the adaptive hash index.

Name	InnoDB Adaptive Hash Index Memory
Range Label	MB
Series	**Expression**
Hash Node Size	`( hash_size * 16384 ) / 1024 / 1024`
Variables	**Data Item**
hash_size	mysql:innodbstatus:innodb_hash_node_heap

InnoDB Adaptive Hash Index Searches

Displays the number of InnoDB adaptive hash index searches and those searches not using the adaptive hash index. You can use this to determine if your adaptive hash index is being used correctly, and is of a suitable size of the information and tables that are being accessed.

Name	InnoDB Adaptive Hash Index Searches
Range Label	searches/sec
Series	**Expression**
hash searches	`hash`

non-hash searches	`nonhash`
Variables	**Data Item**
hash	`mysql:innodbstatus:innodb_hash_searches_per_sec`
nonhash	`mysql:innodbstatus:innodb_non_hash_searches_per_sec`

InnoDB Buffer Pool

Displays the size and page usage of the InnoDB buffer pool, providing information on the cache usage and performance. The graph shows the buffer pool size, active pages, and the number of modified pages.

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**
size	`mysql:innodbstatus:innodb_bp_size`
used	`mysql:innodbstatus:innodb_bp_db_pages`
modified	`mysql:innodbstatus:innodb_bp_modified_pages`

InnoDB Compression Time

Shows the time spent by InnoDB compressing and uncompressing data in compressed tables.

Name	InnoDB Compression Time
Range Label	secs per min
Series	**Expression**
Compressing	`compress`
Uncompressing	`uncompress`
Variables	**Data Item**
compress	`mysql:innodb_compression_time:compress_seconds`
uncompress	`mysql:innodb_compression_time:uncompress_seconds`

InnoDB OS File Access

Displays the InnoDB I/O counts showing the number of `fsync`, read, and write operations used to support InnoDB tables.

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

Shows the row counts per minute for individual SQL operations (READ, INSERT, UPDATE and DELETE).

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	(updated/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

Displays the InnoDB semaphore status. The graph indicates the amount of time that different InnoDB threads have spent waiting due to the locks used to prevent the same structures being updated at the same time. A large number of threads waiting for the semaphores may be a result of disk I/O or connection problems inside InnoDB. You may want to modify your InnoDB thread concurrency.

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

InnoDB Transaction History

The size of the InnoDB transaction history. A high number may indicate a large number of open (uncommitted) transactions.

Name	InnoDB Transaction History
Range Label	Undo Entries
Series	**Expression**
Undo Log Size	hist_len
Variables	**Data Item**
hist_len	mysql:innodbstatus:innodb_trx_history_list_length

InnoDB Transactions

The number of blocked, locked, and active transactions. You can use this to determine the contention on tables for InnoDB transactions.

Name	InnoDB Transactions

Range Label	total
Series	**Expression**
Blocked	`blocked_trx`
Lock Waiting	`lock_trx`
Running	`active_trx`
Variables	**Data Item**
active_trx	`mysql:innodb_active_transactions:count`
blocked_trx	`mysql:innodb_blocked_transactions:count`
lock_trx	`mysql:innodb_lock_wait_transactions:count`

KBytes In/Out

Displays the total Kilobytes per minute of bytes transferred to/from the server by client applications. Spikes in this output may indicate an unusual application operation or connection.

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

The load average of the server hosting the MySQL server. The load average shows the number of processes using or waiting for CPU time in the last 1, 5 and 15 minutes. Load averages showing a load highe than the number of CPU cores may indicate an overloaded server.

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

Displays the current amount of memory used by the agent, and by the Lua component of the 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

Displays the RAM usage on the server for the monitored MySQL instance. You should compare the total and used RAM values to ensure that you are not exceeding your available RAM, which will lead to swapping and performance issues.

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

Displays the use of swap space on the server for the monitored MySQL instance. High swap usage may indicate that your server needs more RAM or that your MySQL configuration needs to be modified, as high levels of swap will have a negative impact on performance.

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

MyISAM Key Buffer Usage

Displays the efficiency of the MyISAM key index buffer usage. On a system using a lot of MyISAM tables, low usage indicates that your MyISAM key buffer size is too high (reducing the RAM available for other buffering). High usage may indicate that you need to increase your MyISAM key buffer size to help improve performance.

Name	MyISAM Key Buffer Usage
Range Label	MB
Series	**Expression**
Modified	(blk_size*not_flushed)/1024/1024
Total Size	key_buffer/1024/1024
Used	(key_buffer - (blk_size*unused))/1024/1024
Variables	**Data Item**
key_buffer	mysql:variables:key_buffer_size
blk_size	mysql:variables:key_cache_block_size

unused	`mysql:status:Key_blocks_unused`
not_flushed	`mysql:status:Key_blocks_not_flushed`

Opened Tables

The number of tables in the open state per minute.

Name	Opened Tables
Range Label	total/min
Series	**Expression**
Opened Tables	`openedTables`
Variables	**Data Item**
openedTables	`mysql:status:Opened_tables`

Query Cache Blocks

The block usage of the query cache. Low usage may indicate that you are not getting the best performance out of your query cache. High numbers may indicate that you need to increase your query cache size to allow more queries to be cached.

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

Displays the hits, inserts and queries not cacheable in the query cache. Low cache hits may indicate that your queries are being expired from the cache before they can be used, which may mean they are unsuitable for storing in the query cache.

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

Displays the number of queries removed from the cache because the size of the query cache was not large enough to store the queries that can be cached. Try increasing your query cache size.

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

Displays the free space and total size of the query cache.

Name	Query Cache Memory
Range Label	MB
Series	**Expression**
Free MB	`free/(1024*1024)`
Size MB	`size/(1024*1024)`
Variables	**Data Item**
free	`mysql`:`status`:`Qcache_free_memory`
size	`mysql`:`variables`:`query_cache_size`

Query Cache Queries

Shows the number of queries stored in the query cache. In normal operation this should be a relatively constant figure. A large number of switches between high and low numbers may indicate that differently sized queries are being inserted into the query cache, and then later removed as a high number of smaller queries are added to the cache. Try increasing the size of your query cache.

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

Displays the number of seconds behind the master for a given slave in a replication scenario. An increasing value means that your slave is unable to keep up with your master.

Name	Replication Delay
Range Label	total seconds
Series	**Expression**
Seconds Behind Master	`sbehind`
Variables	**Data Item**
sbehind	`mysql`:`slavestatus`:`Seconds_Behind_Master`

Row Access Statistics

Shows the average of the number of accesses for data for different query operations for all table reads, regardless of the storage engine being using, against the number of queries being executed.

Name	Row Access Statistics

Range Label	rows
Series	**Expression**
Average Rows Per Query	`((first+key+next+prev+hread_rnd+hread_rnd_next+sort_rows) / questions)`
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 Accesses

Displays the aggregated row access statistics per minute. Information is shown both in terms of the full table scans (which are expensive to perform), and index based accesses.

Name	Row Accesses
Range Label	avg rows/sec
Series	**Expression**
Rows Read via Full Scan	`((hread_rnd+hread_rnd_next)/60)`
Rows Read via Indexes	`((first+key+next+prev)/60)`
Variables	**Data Item**
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`
first	`mysql:status:Handler_read_first`

Row Writes

Shows the number of delete, write (`INSERT`), and update operations on rows per minute on all tables regardless of storage engine.

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

Shows the number of different sort operations performed on queries.

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 Lock Wait Ratio

Shows the percentage of table locks that have had to wait before an operation can continue. High lock ratios can indicate high contention in your tables, with too many operations having to wait, affecting the performance.

Name	Table Lock Wait Ratio
Range Label	lock wait %
Series	**Expression**
Wait Ratio	`((lock_waits/lock_immediate)*100)`
Variables	**Data Item**
lock_waits	`mysql:status:Table_locks_waited`
lock_immediate	`mysql:status:Table_locks_immediate`

Table Locks

Shows the average number of table locks per second.

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

Displays the number of memory temporary tables and disk temporary tables. Disk temporary tables are slower to create, populate, and read back. You may want to increase your memory temporary table size or check your queries to determine whether the use of temporary tables can be minimized.

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

Appendix C. MySQL Enterprise Monitor Component Reference

C.1. MySQL Enterprise Service Manager Reference

C.1.1. 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:

Description	**Parameter**	**Default Value**
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 should 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 are running the MySQL repository and the Tomcat service on the same machine, you should keep take the memory allocated to MySQL into consideration when setting these values.

C.1.2. 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.

C.1.3. 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.

All log files except catalina.out are rotated to ensure they do not grow beyond 10MB in size.

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 2.8.6, “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.1.4. 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.1.5. 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.

C.2. MySQL Enterprise Agent Reference

The MySQL Enterprise Agent is responsible for providing all of the information from a monitor MySQL server to MySQL Enterprise Service Manager. In addition, the MySQL Enterprise Agent may provide MySQL Enterprise Agent Proxy Service functionality for providing query analyzer data.

The MySQL Enterprise Agent is configured through files located within the `etc` directory within agent installation directory.

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 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 C.2.1, “MySQL Enterprise Agent

Configurable Options”. For details on the content of the individual MySQL instance configuration files, see Section C.2.2, “MySQL Server (`agent-instance.ini`) Configuration”.

C.2.1. MySQL Enterprise Agent Configurable Options

The `mysql-monitor-agent.ini` files contains the base configuration information for the MySQL Enterprise 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 and adjusting other parameters as necessary.
#
# WARNING - If you are upgrading from a previous 1.x release, please
# note that we have changed the contents of the agent INI file and
# split the contents across two files:
#
#   <installdir>/etc/mysql-monitor-agent.ini              - this file
#   <installdir>/etc/instances/mysql/agent-instance.ini   - new file
#
# Refer to the documentation for more detailed information and
# instructions.
#
# Version: 2.2.0.1708

[mysql-proxy]

# Common Parameters
plugins=proxy,agent
keepalive = true

log-file = mysql-monitor-agent.log
pid-file=/opt/mysql/enterprise/agent/mysql-monitor-agent.pid

# Agent Parameters
agent-mgmt-hostname = http://nautilus:18080/heartbeat
agent-mgmt-username = agent
agent-mgmt-password = Tamsin
mysqld-instance-dir= etc/instances
agent-item-files = share/mysql-monitor-agent/items/quan.lua,share/mysql-monitor-agent/items/items-mysql-monitor.xml,share/
agent/items/custom.xml
agent-uuid = 68b4892c-1310-4f07-8924-3a8b3c0e850b
user = root

# Proxy Parameters
proxy-address=:6446
proxy-backend-addresses = 127.0.0.1:3306
proxy-lua-script = lib/mysql-monitor-agent/lua/quan.lua
```

> **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:

The MySQL Enterprise Agent supports the following configurable options:

Table C.6. `mysql-monitor-agent` Help Options

Format	Option File	Description	Introduction	Deprecated	Removed
--help		Show help options			
--help-admin		Show options for the admin-module			
--help-agent		Show agent options			
--help-all		Show all help options			
--help-monitor		Show monitor options			
--help-proxy		Show options for the proxy-module			

Table C.7. mysql-monitor-agent Admin Options

Format	Option File	Description	Introduction	Deprecated	Removed
--admin-address	admin-address	Defines the listening address and port for the admin module			
--admin-lua-script	admin-lua-script	Script to execute by the admin module			
--admin-password	admin-password	Password for authentication for admin module			
--admin-username	admin-username	Username for authentication for admin module			
--proxy-address	proxy-address	Listening address:port of the proxy server			

Table C.8. mysql-monitor-agent Agent Options

Format	Option File	Description	Introduction	Deprecated	Removed
--agent-backlog-timeout	agent-backlog-timeout	Maximum time in seconds that the Set the second after the minute when the data is sent to the server	2.2.0		
--agent-collector-plugins	agent-collector-plugins	Load the specified collector plugins			
--agent-connect-timeout	agent-connect-timeout	The timeout for the agent to connect to the MySQL server	2.2.2		
--agent-DNS-cachetimeout	agent-DNS-cachetimeout	The timeout for cache operations	2.2.0		
--agent-force-uncorking	agent-force-uncorking	Forcibly turn off corking (randomization of data collection intervals)			
--agent-generate-uuid	agent-generate-uuid	Generate a UUID for use with the agent-uuid			
--agent-heartbeat-interval	agent-heartbeat-interval	The interval for generating heartbeat operations			
--agent-host-id	agent-host-id	The host ID for the agent			
--agent-host-id-commandline	agent-host-id-commandline	The command to use to generate the agent host id			
--agent-hostname	agent-hostname	Hostname of the agent host			
--agent-IP-version	agent-IP-version	IP version (4 or 6) to force	2.2.0		
--agent-item-files	agent-item-files	List of data items for additional collections			
--	agent-master-uuid-discovery	Disable the use of the master_uuid for topology discovery	2.2.1		

Format	Option File	Description	Introduction	Deprecated	Removed
agent-master-uuid-discovery					
--agent-max-response-size	agent-max-response-size	Maximum response size for agent data			
--agent-mgmt-hostname	agent-mgmt-hostname	URL of the management server			
--agent-mgmt-password	agent-mgmt-password	Password for the management server	2.2.0		
--agent-mgmt-username	agent-mgmt-username	User name with the management server	2.2.0		
--agent-net-timeout	agent-net-timeout	The timeout for the agent for established connections to the MySQL server	2.2.2		
--agent-report-network-fs	agent-report-network-fs	Enable data collection for networked filesystems	2.2.0		
--agent-report-offset	agent-report-offset	Set the second after the minute when the data is sent to the server	2.2.0		
--agent-run-os-tests	agent-run-os-tests	Run the operating system tests and shutdown			
--agent-sync-attempts	agent-sync-attempts	Synchronize the attempts to connect at resync			
--agent-track-alloc	agent-track-alloc	Enable the tracking of the allocation sizes			
--agent-uuid	agent-uuid	UUID of this agent			
--mysqld-instance-dir	mysqld-instance-dir	Location of the MySQL instance configuration (mysql.cnf/mysql.ini) files	2.2.0		

Table C.9. mysql-monitor-agent Monitor Options

Format	Option File	Description	Introduction	Deprecated	Removed
--monitor-interval	monitor-interval	Interval for executing queries against the backend			
--monitor-lua-script	monitor-lua-script	Script filename for the monitor			
--monitor-password	monitor-password	Set the password for the monitored MySQL Server			
--monitor-username	monitor-username	Set the username for the monitored MySQL Server			

Table C.10. `mysql-monitor-agent` Proxy Options

Format	Option File	Description	Introduction	Deprecated	Removed
--no-proxy	no-proxy	Don't start the proxy module			
--proxy-backend-addresses	proxy-backend-addresses	Address:port of the remote MySQL server			
--proxy-fix-bug-25371	proxy-fix-bug-25371	Enable the fix for Bug#25371 (for mysqld > 2.0.0.12) for older libmysql versions			
--proxy-lua-script	proxy-lua-script	Filename for Lua script for proxy operations			
--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			
--proxy-read-only-backend-addresses	proxy-read-only-backend-addresses	Address:port of the remote MySQL server (read-only)			
--proxy-skip-profiling	proxy-skip-profiling	Disabled profiling of queries			

Table C.11. `mysql-monitor-agent` Application Options

Format	Option File	Description	Introduction	Deprecated	Removed
--basedir	basedir	Specify the base directory to prepend to paths in the config			
--daemon	daemon	Start in daemon-mode			
--defaults-file	defaults-file	Specify the configuration file			
--keepalive	keepalive	Try to restart the proxy if a crash occurs			
--log-backtrace-on-crash	log-backtrace-on-crash	Try to invoke the debug and generate a backtrace on crash			
--log-file	log-file	Specify the file for logging error messages			
--log-level	log-level	Logging level			
--log-use-syslog	log-use-syslog	Log errors to syslog			
--lua-cpath	lua-cpath	Set the LUA_CPATH			
--lua-path	lua-path	Set the LUA_PATH			
--max-open-files	max-open-files	Specify the maximum number of open files to support			
--pid-file	pid-file	PID file to store the process ID (when in daemon mode)			
--plugin-dir	plugin-dir	Path to the plugin files			

Format	Option File	Description	Introduction	Deprecated	Removed
--plugins	plugins	List of plugins to load			
--user	user	Specify the user to use when running mysql-monitor-agent			
--verbose-shutdown	verbose-shutdown	Report shutdown in the logs regardless of log level	2.2.0		
--version	version	Show the version information			

- `--help`

Command-Line Format	`--help`
	`-?`

  Show available help options.

- `--help-all`

Command-Line Format	`--help-all`

  Show all help options.

- `--help-admin`

Command-Line Format	`--help-admin`

  Show options for the admin-module.

- `--help-proxy`

Command-Line Format	`--help-proxy`

  Show options for the proxy-module.

- `--help-monitor`

Command-Line Format	`--help-monitor`

  Show options for the monitor module.

- `--help-agent`

Command-Line Format	`--help-agent`

  Show options for the agent module.

- `--admin-address=host:port`

Command-Line Format	`--admin-address`
Option-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

Command-Line Format	--admin-lua-script	
Option-File Format	admin-lua-script	
	Permitted Values	
	Type	file name
	Default	

Specify the script to use for the administration module for the proxy.

- --admin-password=pass

Command-Line Format	--admin-password	
Option-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

Command-Line Format	--admin-username	
Option-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

Command-Line Format	--agent-collector-plugins	
Option-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.

When monitoring servers you must have the agent plugin configured. To support Query Analyzer, you must also have the proxy module enabled. Plugins should be specified as a comma separated list of plugin names.

If you enabled Query Analyzer during installation of the agent, the default value is `proxy,agent`. If you disabled Query Analyzer during installation, the default value is `agent`.

- `--agent-connect-timeout=seconds`

Version Introduced	2.2.2	
Command-Line Format	`--agent-connect-timeout`	
Option-File Format	`agent-connect-timeout`	
	Permitted Values	
	Type	`numeric`
	Default	`10`

The timeout for the agent to connect to the MySQL server before a failure is reported.

- `--agent-DNS-cachetimeout=seconds`

Version Introduced	2.2.0		
Command-Line Format	`--agent-DNS-cachetimeout`		
Option-File Format	`agent-DNS-cachetimeout`		
	Permitted Values		
	Type	`numeric`	
	Default	`0`	
	Valid Values	`-1`	Never timeout
		`0`	Don't cache

The timeout for caching the IP address of a given DNS name. By default, the operating system caching of the address is used and no caching in the agent is performed at all. If you specify -1 caching is enabled and never times out. Any other positive value specifies the timeout value.

- `--agent-net-timeout=seconds`

Version Introduced	2.2.2	
Command-Line Format	`--agent-net-timeout`	
Option-File Format	`agent-net-timeout`	
	Permitted Values	
	Type	`numeric`
	Default	`10`

The timeout for the agent to get a response from a MySQL server to which it is already connected before the failure is reported.

- `--agent-report-network-fs`

Version Introduced	2.2.0	
Command-Line Format	`--agent-report-network-fs`	
Option-File Format	`agent-report-network-fs`	
	Permitted Values	
	Type	`boolean`
	Default	`off`

Enables data collection for network filesystems. In MySQL Enterprise Agent 2.2.2 and greater, the reporting of network filesystem information is disabled because the latency in determining and reporting the information can delay the collection and reporting of other information. You must manually enable collection if you need or use advisor and graphs that perform custom data collection on networked filesystems.

- `--agent-force-uncorking`

Command-Line Format	`--agent-force-uncorking`
Option-File Format	`agent-force-uncorking`

Forcibly turn off the corking of responses sent to MySQL Enterprise Service Manager. The corking mechanism tries to randomize the reporting interval of data sent up to the server to prevent flooding the server with information.

You can combine this option with the `---agent-report-offset` to specify the offset when data is reported to the server. For more information, see `agent-report-offset`.

The frequency of collecting the data is not modified by this setting.

- `--agent-generate-uuid`

Command-Line Format	`--agent-generate-uuid`
Option-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 5.3.6, “Monitoring Multiple MySQL Servers”.

- `--agent-heartbeat-interval=#`

Command-Line Format	`--agent-heartbeat-interval`	
Option-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 alive 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`

Command-Line Format	`--agent-host-id`
Option-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.

- `--agent-host-id=HOSTID`

Version Introduced	2.2.0	
Command-Line Format	`--agent-IP-version`	
Option-File Format	`agent-IP-version`	
	Permitted Values	
	Type	`numeric`
	Default	`any`
	Valid Values	`4`, `6`

Specify the IP version to be used and supported by MySQL Enterprise Agent. By default, the agent supports both IPv4 and IPv6, or you can specify the version by specifying `4` or `6`.

- `--host-id-commandline`

Command-Line Format	`--agent-host-id-commandline`
Option-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

- `--agent-hostname`

Command-Line Format	`--agent-hostname`
Option-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.

Note

Setting the hostname using the **Manage Servers** tab within MySQL Enterprise Dashboard overrides this setting.

- `--agent-item-files`

Command-Line Format	`--agent-item-files`
Option-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 4.1.8, “Creating a Custom Data Collection Item”.

The default value includes `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).

- `--agent-max-response-size`

Command-Line Format	`--agent-max-response-size`	
Option-File Format	`agent-max-response-size`	
	Permitted Values	
	Type	`numeric`
	Default	`65536`

Specify the maximum size of the response packet sent to MySQL Enterprise Service Manager

- `--agent-master-uuid-discovery`

Version Introduced	2.2.1	
Command-Line Format	`--agent-master-uuid-discovery`	
Option-File Format	`agent-master-uuid-discovery`	
	Permitted Values	
	Type	`boolean`
	Default	`true`

By default, the MySQL Enterprise Agent uses the UUID of the replication master to aid in identifying the replication topology. Setting this option to false prevents the agent from accessing the `master.info` and using the UUID obtained from the replication master. Instead, the hostname and port information is used to determine the topology and match the instance and master information.

- `--agent-mgmt-hostname`

Command-Line Format	`--agent-mgmt-hostname`
Option-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-mgmt-username`

Version Introduced	2.2.0	
Command-Line Format	`--agent-mgmt-username`	
Option-File Format	`agent-mgmt-username`	
	Permitted Values	
	Type	`string`

The username to be used when communicating with MySQL Enterprise Service Manager.

- `--agent-mgmt-password`

Version Introduced	2.2.0	
Command-Line Format	`--agent-mgmt-password`	
Option-File Format	`agent-mgmt-password`	
	Permitted Values	
	Type	`string`

The password to be used when communicating with MySQL Enterprise Service Manager.

- `--agent-report-offset`

Version Introduced	2.2.0

Command-Line Format	`--agent-report-offset`	
Option-File Format	`agent-report-offset`	
	Permitted Values	
	Type	`numeric`
	Default	`random`

The offset, in seconds, past the minute when data is sent up to MySQL Enterprise Service Manager. You can use this to harmonize or synchronize agent report across multiple machines, which can improve performance.

For example, if you set `--agent-report-offset=15`, then data is reported at 15 seconds past each minute. For example, at 12:00:15, 12:01:15, and so on.

- `--agent-run-os-tests`

Command-Line Format	`--agent-run-os-tests`
Option-File Format	`agent-run-os-tests`

Runs the internal operating system tests, and then exits. Using this option generates 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 2.2.1.1138 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`

Command-Line Format	`--agent-sync-attempts`
Option-File Format	`agent-sync-attempts`

Attempt to synchronize with the server during the resynchronization.

- `--track-alloc`

Command-Line Format	`--agent-track-alloc`
Option-File Format	`agent-track-alloc`

Track the memory allocation in the various modules of the agent to help monitor the memory usage.

- `--agent-uuid`

Command-Line Format	`--agent-uuid`	
Option-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.

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 leaving this configuration property blank.

- `--basedir`

Command-Line Format	`--basedir`	
Option-File Format	`basedir`	
	Permitted Values	
	Type	`file name`

Specify a base directory that is prepended to all other filename configuration options. Use an absolute (not relative) directory name, otherwise `mysql-monitor-agent` generates an error during startup.

- `--defaults-file`

Command-Line Format	`--defaults-file`
Option-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`	
Option-File Format	`event-threads`	
	Permitted Values	
	Type	`numeric`
	Default	`1`

Specify the number of event threads reserved to handle incoming requests.

- `--keepalive`

Command-Line Format	`--keepalive`
Option-File Format	`keepalive`

Creates an angel process surrounding the main `mysql-monitor-agent` that attempts to restart the true `mysql-monitor-agent` process in the event a crash or other failure.

If the agent stops unexpectedly, the angel revives the agent.

An expected shutdown is:

- MySQL Enterprise Agent terminates normally
- Shutdown with `SIGTERM`
- Startup errors

If the angel recieves one of `SIGHIP`, `SIGTERM`, or `SIGINT` signals, the signal is forwarded to the agent to initiate shutdown.

- `--agent-backlog-timeout=VALUE`

Version Introduced	2.2.0	
Command-Line Format	`--agent-backlog-timeout`	
Option-File Format	`agent-backlog-timeout`	
	Permitted Values	
	Type	`numeric`

The time in seconds that data is kept by the agent if the agent is unable to communicate with MySQL Enterprise Service Manager. Once the agent reconnects, the data is sent up to MySQL Enterprise Service Manager. Data older than the specified value is discarded. The default value is 600 seconds.

- `--log-backtrace-on-crash`

Command-Line Format	`--log-backtrace-on-crash`
Option-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`

Command-Line Format	`--log-file`	
Option-File Format	`log-file`	
	Permitted Values	
	Type	`file name`

Specify the name of a file to be used to record log information. 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.

- `--log-file=filename`

Command-Line Format	`--log-level`
Option-File Format	`log-level`

	Permitted Values		
	Type	enumeration	
	Default	critical	
	Valid Values	error	Show error messages
		warning	Show only warning messages
		info	Show informational messages
		message	Show information about agent and basic processing
		critical	Show critical messages highlighting agent problems
		debug	Show detailed information, including info provided to server

The log level used when outputting error messages. Messages of that level (or lower) are output. For example, `message` also outputs `info`, `warning`, and `error` messages.

> ### Warning
>
> Be careful when setting the `log-level` to `debug`. Doing this rapidly increases 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.

- `--log-use-syslog`

Command-Line Format	`--log-use-syslog`
Option-File Format	`log-use-syslog`

Causes errors to be sent to the syslog (Unix/Linux only).

- `--lua-cpath=path`

Command-Line Format	`--lua-cpath`	
Option-File Format	`lua-cpath`	
	Permitted Values	
	Type	file name

Specify the `LUA_CPATH` to be used when loading compiled modules or libraries for Lua scripts.

- `--lua-path=path`

Command-Line Format	`--lua-path`	
Option-File Format	`lua-path`	
	Permitted Values	
	Type	file name

Specify the `LUA_CPATH` to be used when loading modules for Lua.

- `--max-open-files=#`

Command-Line Format	`--max-open-files`	
Option-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=#`

Command-Line Format	`--monitor-interval`	
Option-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`

Command-Line Format	`--monitor-lua-script`	
Option-File Format	`monitor-lua-script`	
	Permitted Values	
	Type	`string`

  Filename of the Lua script to use for the monitor module.

- `--monitor-password=PASS`

Command-Line Format	`--monitor-password`	
Option-File Format	`monitor-password`	
	Permitted Values	
	Type	`string`

  The password for the monitoring user account.

- `--monitor-username=USER`

Command-Line Format	`--monitor-username`	
Option-File Format	`monitor-username`	
	Permitted Values	
	Type	`string`
	Default	`monitor`

  The username for the monitoring user account.

- `--mysqld-instance-dir=PATH`

Version Introduced	2.2.0	
Command-Line Format	`--mysqld-instance-dir`	
Option-File Format	`mysqld-instance-dir`	
	Permitted Values	
	Type	`string`

The path to the configuration files for the MySQL instance.

- `--agent-IP-version=VER`

Version Introduced	2.2.0	
Command-Line Format	`--agent-IP-version`	
Option-File Format	`agent-IP-version`	
	Permitted Values	
	Type	`numeric`
	Default	`any`
	Valid Values	`4`, `6`

Explicitly sets the IP version to be used for addressing. By default, the agent uses either version. Set the value by specifying `4` (for IPv4) or `6` (for IPv6).

- `--no-proxy`

Command-Line Format	`--no-proxy`
Option-File Format	`no-proxy`

Disable the proxy module.

- `--plugin-dir=path`

Command-Line Format	`--plugin-dir`	
Option-File Format	`plugin-dir`	
	Permitted Values	
	Type	`file name`

The directory to use when loading plugins for `mysql-monitor-agent`.

- `--plugins=plugin,...`

Command-Line Format	`--plugins`	
Option-File Format	`plugins`	
	Permitted Values	
	Type	`string`

A comma-separated list of the plugins to be loaded.

- `--proxy-address=host:port`

Command-Line Format	`--proxy-address`

	`-P`	
Option-File Format	`proxy-address`	
	Permitted Values	
	Type	`string`
	Default	`:6446`

The listening host name (or IP address) and port of the proxy service. The default is `:6446` (all IPs on port 6446). 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.

- `--proxy-read-only-backend-addresses=host:port`

Command-Line Format	`--proxy-read-only-backend-addresses`	
	`-r`	
Option-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 goes to server A; the second client connection goes to server B; and the third client connection goes to server A.

- `--proxy-pool-no-change-user`

Command-Line Format	`--proxy-pool-no-change-user`
Option-File Format	`proxy-pool-no-change-user`

Disables the use of the MySQL protocol `CHANGE_USER` when re-using 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`

Command-Line Format	`--proxy-fix-bug-25371`
Option-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`

Command-Line Format	`--proxy-lua-script`
	`-s`
Option-File Format	`proxy-lua-script`

	Permitted Values	
	Type	`file name`

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, the updated content is automatically used when a new connection is made. To use Query Analyzer, set this parameter to `share/mysql-proxy/quan.lua`. This is the default value.

- `--daemon`

Command-Line Format	`--daemon`
Option-File Format	`daemon`

Starts the proxy in daemon mode.

- `--pid-file=file`

Command-Line Format	`--pid-file`	
Option-File Format	`pid-file`	
	Permitted Values	
	Type	`file name`

Sets the name of the file to be used to store the process ID. 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

- `--user=user`

Command-Line Format	`--user`	
Option-File Format	`user`	
	Permitted Values	
	Type	`string`

Run `mysql-monitor-agent` as the specified `user`.

- `--version`

Command-Line Format	`--version`
	`-V`
Option-File Format	`version`

Show the version number.

- `--verbose-shutdown`

Version Introduced	2.2.0	
Command-Line Format	`--verbose-shutdown`	
Option-File Format	`verbose-shutdown`	
	Permitted Values	
	Type	`boolean`
	Default	`off`

By default, agent shutdown messages are only logged if you have set `log-level` to `message` or higher. You can force shutdown messages to be reported regardless of the log level using this option.

C.2.2. 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
```

Replication topology discovery for an instance is determined by the agent using the `master.info` file for the instance, extracting the username/password used to connect to the master, and then obtaining the MySQL Enterprise Monitor generated UUID of the replication master. This enables the MySQL Enterprise Agent to match the slave and master in the replication topology.

You can provide alternative credentials to be used to obtain the master UUID by adding a `replication` section to the instance configuration file, and providing values for the `discovery-username` and `discovery-password` options. For example:

```
[replication]
discovery-username=agentdiscovery
discovery-password=password
```

To switch off the use of the master UUID entirely when determining the replication topology, use the `agent-master-uuid-discovery` option.

C.2.3. 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 Agent cannot add information to the configured logfile, then information is sent to the standard output instead.

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 rotates the log before it reaches 1GB in size, keeping up to 5 files at a time.

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.

Appendix D. 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 2.3.3, “Editing Built-in Rules” and Section 4.1.2, “Overview of Rule Creation”.

Table D.1. MySQL Monitor Data Items

Namespace	Namespace Type	Attribute	Type	Description
agent	allocationstats	agent_dataitem_spec_alloc	long_counter	MISSING
agent	allocationstats	agent_dataitem_spec_free	long_counter	MISSING
agent	allocationstats	agent_item_attribute_alloc	long_counter	MISSING
agent	allocationstats	agent_item_attribute_free	long_counter	MISSING
agent	allocationstats	agent_item_class_alloc	long_counter	MISSING
agent	allocationstats	agent_item_class_config_alloc	long_counter	MISSING
agent	allocationstats	agent_item_class_config_free	long_counter	MISSING
agent	allocationstats	agent_item_class_free	long_counter	MISSING
agent	allocationstats	agent_item_classes_alloc	long_counter	MISSING
agent	allocationstats	agent_item_classes_free	long_counter	MISSING
agent	allocationstats	agent_item_instance_alloc	long_counter	MISSING
agent	allocationstats	agent_item_instance_free	long_counter	MISSING
agent	allocationstats	agent_item_instance_name_alloc	long_counter	MISSING
agent	allocationstats	agent_item_instance_name_free	long_counter	MISSING
agent	allocationstats	agent_item_namespaces_alloc	long_counter	MISSING
agent	allocationstats	agent_item_namespaces_free	long_counter	MISSING
agent	allocationstats	agent_item_value_alloc	long_counter	MISSING
agent	allocationstats	agent_item_value_free	long_counter	MISSING
agent	allocationstats	agent_mysqld_alloc	long_counter	MISSING
agent	allocationstats	agent_mysqld_free	long_counter	MISSING
agent	allocationstats	agent_plugin_alloc	long_counter	MISSING
agent	allocationstats	agent_plugin_free	long_counter	MISSING
agent	allocationstats	agent_target_alloc	long_counter	MISSING

Namespace	Namespace Type	Attribute	Type	Description
agent	allocationstats	agent_target_free	long_counter	MISSING
agent	allocationstats	agent_task_alloc	long_counter	MISSING
agent	allocationstats	agent_task_free	long_counter	MISSING
agent	allocationstats	curl_mem_alloc	long_counter	MISSING
agent	allocationstats	curl_mem_bytes	long	MISSING
agent	allocationstats	curl_mem_bytes_max	long	MISSING
agent	allocationstats	curl_mem_free	long_counter	MISSING
agent	allocationstats	job_response_alloc	long_counter	MISSING
agent	allocationstats	job_response_data_alloc	long_counter	MISSING
agent	allocationstats	job_response_data_free	long_counter	MISSING
agent	allocationstats	job_response_datum_alloc	long_counter	MISSING
agent	allocationstats	job_response_datum_free	long_counter	MISSING
agent	allocationstats	job_response_free	long_counter	MISSING
agent	allocationstats	job_response_string_alloc	long_counter	MISSING
agent	allocationstats	job_response_string_free	long_counter	MISSING
agent	allocationstats	job_task_alloc	long_counter	MISSING
agent	allocationstats	job_task_free	long_counter	MISSING
agent	allocationstats	network_backlog_alloc	long_counter	MISSING
agent	allocationstats	network_backlog_free	long_counter	MISSING
agent	allocationstats	network_backlog_node_alloc	long_counter	MISSING
agent	allocationstats	network_backlog_node_bytes	long	MISSING
agent	allocationstats	network_backlog_node_free	long_counter	MISSING
agent	allocationstats	scheduler_job_alloc	long_counter	MISSING
agent	allocationstats	scheduler_job_free	long_counter	MISSING
agent	allocationstats	xml_mem_alloc	long_counter	MISSING
agent	allocationstats	xml_mem_bytes	long	MISSING
agent	allocationstats	xml_mem_bytes_max	long	MISSING
agent	allocationstats	xml_mem_free	long_counter	MISSING
agent	chassisstats	glib_mem_alloc	long_counter	MISSING
agent	chassisstats	glib_mem_bytes	long	MISSING
agent	chassisstats	glib_mem_bytes_max	long	MISSING
agent	chassisstats	glib_mem_free	long_counter	MISSING
agent	chassisstats	lua_mem_alloc	long_counter	MISSING
agent	chassisstats	lua_mem_bytes	long	MISSING
agent	chassisstats	lua_mem_bytes_max	long	MISSING
agent	chassisstats	lua_mem_free	long_counter	MISSING

Namespace	Namespace Type	Attribute	Type	Description
agent	lua	mem_size	long	MISSING
agent	proc	egid	long	MISSING
agent	proc	euid	long	MISSING
agent	proc	gid	long	MISSING
agent	proc	mem_major_faults	long_counter	MISSING
agent	proc	mem_minor_faults	long_counter	MISSING
agent	proc	mem_page_faults	long_counter	MISSING
agent	proc	mem_resident	long	MISSING
agent	proc	mem_share	long	MISSING
agent	proc	mem_size	long	MISSING
agent	proc	name	string	MISSING
agent	proc	nice	long	MISSING
agent	proc	ppid	long	MISSING
agent	proc	priority	long	MISSING
agent	proc	processor	long	MISSING
agent	proc	state	long	MISSING
agent	proc	threads	long	MISSING
agent	proc	time_start_time	long	MISSING
agent	proc	time_sys	long_counter	MISSING
agent	proc	time_total	long_counter	MISSING
agent	proc	time_user	long_counter	MISSING
agent	proc	tty	long	MISSING
agent	proc	uid	long	MISSING
monitor	resourceBundle	subscription	string	MISSING
monitor	resourceBundle	version	string	MISSING
mysql	account_old_password	user	string	MISSING
mysql	active_count	active_count	long	MISSING
mysql	Agent	agent.latency	long	MISSING
mysql	Agent	agent.reachable	string	MISSING
mysql	Agent	host_id	string	MISSING
mysql	Agent	name	string	MISSING
mysql	Agent	quanDataOutOfBand	string	MISSING
mysql	Agent	version	string	MISSING
mysql	anonymous_user	user_count	long	MISSING
mysql	auto_increment_limit	table_list	string	MISSING
mysql	avail_count	avail_count	long	MISSING
mysql	broad_host_specifier	user	string	MISSING
mysql	cluster_data_node_data_memory	data_mem_total	long	MISSING
mysql	cluster_data_node_data_memory	data_mem_used	long	MISSING
mysql	cluster_data_node_index_memory	index_mem_total	long	MISSING
mysql	cluster_data_node_index_memory	index_mem_used	long	MISSING
mysql	cluster_data_node_redo_	redo_buffer_total	long	MISSING

Namespace	Namespace Type	Attribute	Type	Description
	buffer			
mysql	cluster_data_node_redo_buffer	redo_buffer_used	long	MISSING
mysql	cluster_data_node_redo_logspace	redo_logspace_total	long	MISSING
mysql	cluster_data_node_redo_logspace	redo_logspace_used	long	MISSING
mysql	cluster_data_node_undo_buffer	undo_buffer_total	long	MISSING
mysql	cluster_data_node_undo_buffer	undo_buffer_used	long	MISSING
mysql	cluster_data_node_undo_logspace	undo_logspace_total	long	MISSING
mysql	cluster_data_node_undo_logspace	undo_logspace_used	long	MISSING
mysql	cluster_data_nodes_not_started	node_list	string	MISSING
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	MISSING
mysql	Database	Database	string	MISSING
mysql	Database	name	string	MISSING
mysql	Explain	extra	string	MISSING
mysql	Explain	id	long	MISSING
mysql	Explain	key	string	MISSING
mysql	Explain	key_len	string	MISSING
mysql	Explain	possible_keys	string	MISSING
mysql	Explain	ref	string	MISSING
mysql	Explain	rows	long	MISSING
mysql	Explain	select_type	string	MISSING
mysql	Explain	table	string	MISSING
mysql	Explain	type	string	MISSING
mysql	falcon_database_io	logical_reads	long	MISSING
mysql	falcon_database_io	physical_reads	long	MISSING
mysql	falcon_record_cache_summary	total_recordcache_mb	double	MISSING
mysql	falcon_record_cache_summary	total_space_mb	double	MISSING
mysql	falcon_system_memory_summary	free_system_mb	double	MISSING
mysql	fal-	total_system_mb	double	MISSING

Namespace	Namespace Type	Attribute	Type	Description
	con_system_memory_summary			
mysql	falcon_transaction_summary	committed_txns	long	MISSING
mysql	falcon_transaction_summary	rolled_back_txns	long	MISSING
mysql	global_privileges	user	string	MISSING
mysql	grant_privileges	user_spec	string	MISSING
mysql	inappropriate_slave_privileges	user	string	MISSING
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	innodb_active_transactions	count	long	MISSING
mysql	innodb_blocked_transactions	count	long	MISSING
mysql	innodb_compression_time	compress_seconds	long_counter	MISSING
mysql	innodb_compression_time	uncompress_seconds	long_counter	MISSING
mysql	innodb_lock_wait_transactions	count	long	MISSING
mysql	innodb_plugins	innodb_plugin_list	string	MISSING
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	in-	long	MISSING

Namespace	Namespace Type	Attribute	Type	Description
		nodb_bp_dictionary_alloc		
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_io_cur_pages	long	MISSING
mysql	innodbstatus	innodb_bp_io_sum_pages	long	MISSING
mysql	innodbstatus	innodb_bp_io_unzip_cur_pages	long	MISSING
mysql	innodbstatus	innodb_bp_io_unzip_sum_pages	long	MISSING
mysql	innodbstatus	innodb_bp_lru_len	long	MISSING
mysql	innodbstatus	innodb_bp_modified_pages	long	The current number of pages modified
mysql	innodbstatus	innodb_bp_not_young_hit_rate	long	MISSING
mysql	innodbstatus	innodb_bp_old_db_pages	long	MISSING
mysql	innodbstatus	innodb_bp_pages_created	long_counter	The total number of pages created
mysql	innodbstatus	innodb_bp_pages_evicted_no_access_per_sec	double	MISSING
mysql	innodbstatus	innodb_bp_pages_not_young	long_counter	MISSING
mysql	innodbstatus	innodb_bp_pages_not_young_per_sec	double	MISSING
mysql	innodbstatus	innodb_bp_pages_read	long_counter	The total number of pages read
mysql	innodbstatus	innodb_bp_pages_read_ahead_per_sec	double	MISSING
mysql	innodbstatus	innodb_bp_pages_written	long_counter	The total number of pages written
mysql	innodbstatus	innodb_bp_pages_young	long_counter	MISSING
mysql	innodbstatus	innodb_bp_pages_young_per_sec	double	MISSING
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	in-	long	The number of pending independent

Namespace	Namespace Type	Attribute	Type	Description
		nodb_bp_pending_writes_single_page		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_unzip_lru_len	long	MISSING
mysql	innodbstatus	innodb_bp_written_per_sec	double	The number of buffer pool pages written per second
mysql	innodbstatus	innodb_bp_young_hit_rate	long	MISSING
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_free_list_len	long	MISSING
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_ibuf_seg_size	long	MISSING
mysql	innodbstatus	innodb_ibuf_size	long	MISSING
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
mysql	innodbstatus	innodb_io_pending_flush_bp	long	The number of pending buffer pool flush operations

Namespace	Namespace Type	Attribute	Type	Description
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_main_thd_log_flush_writes	long_counter	MISSING
mysql	innodbstatus	innodb_main_thd_loops_background	long_counter	MISSING
mysql	innodbstatus	innodb_main_thd_loops_flush	long_counter	MISSING
mysql	innodbstatus	innodb_main_thd_loops_one_sec	long_counter	MISSING
mysql	innodbstatus	innodb_main_thd_loops_sleeps	long_counter	MISSING
mysql	innodbstatus	innodb_main_thd_loops_ten_sec	long_counter	MISSING
mysql	innodbstatus	innodb_non_hash_searches_per_sec	double	The number of nonadaptive hash index searches

Namespace	Namespace Type	Attribute	Type	Description
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	innodb_rows_deleted_per_sec	double	The number of rows deleted per second
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

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodbstatus	innodb_sem_shared_spins	long_counter	The number of shared (read) semaphore spin waits within the syn array
mysql	innodbstatus	innodb_sem_spins_per_wait_mutex	double	MISSING
mysql	innodbstatus	innodb_sem_spins_per_wait_rw_excl	double	MISSING
mysql	innodbstatus	innodb_sem_spins_per_wait_rw_shared	double	MISSING
mysql	innodbstatus	innodb_trx_history_list_length	long	The transaction history list length
mysql	innodbstatus	innodb_trx_id_counter	long_counter	MISSING
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_id_counter_str	string	MISSING
mysql	innodbstatus	innodb_trx_purge_done_trx2	long	The transaction count that is already purged
mysql	innodbstatus	innodb_trx_total_lock_structs	long	MISSING
mysql	innodbstatus	innodb_writes_per_sec	double	The number of InnoDB writes per second
mysql	locked_processes	num_locked	long	MISSING
mysql	long_locked_processes	long_running_time	double	MISSING
mysql	long_locked_processes	num_long_running	long	MISSING
mysql	long_running_processes	long_running_time	double	MISSING
mysql	long_running_processes	num_long_running	long	MISSING
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	MISSING
mysql	MasterStatus	Binlog_Ignore_DB	string	MISSING
mysql	MasterStatus	File	string	MISSING
mysql	MasterStatus	Position	long	MISSING
mysql	no_password	user	string	MISSING
mysql	no_root_password	no_password	long	MISSING
mysql	num_waiting_txns	num_waiting_txns	long	MISSING
mysql	prepared_count	prepared_count	long	MISSING
mysql	privileges_on_all_dbs	user_spec	string	MISSING
mysql	processlist	Command	string	The type of command the thread is executing

Namespace	Namespace Type	Attribute	Type	Description
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	MISSING
mysql	rolledback_count	rolledback_count	long	MISSING
mysql	root_account_exists	root_account	long	MISSING
mysql	root_remote_login	remote_login	long	MISSING
mysql	Server	blackout	string	MISSING
mysql	Server	displayname	string	The display name of the server in the Dashboard
mysql	Server	registration-complete	string	MISSING
mysql	Server	repl.group	long	MISSING
mysql	Server	repl.groupName	string	MISSING
mysql	Server	server.connected	long	Whether the server is connected
mysql	Server	server.last_errno	string	MISSING
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	MISSING
mysql	Server	transport	string	MISSING
mysql	Server	visible.displayname	string	MISSING
mysql	server_admin_privileges	user_spec	string	MISSING
mysql	SlaveStatus	Binlog_Do_DB	string	MISSING
mysql	SlaveStatus	Binlog_Ignore_DB	string	MISSING
mysql	SlaveStatus	Connect_Retry	long	MISSING
mysql	SlaveStatus	Exec_Master_Log_Pos	long	MISSING
mysql	SlaveStatus	File	string	MISSING
mysql	SlaveStatus	Last_Errno	long	MISSING
mysql	SlaveStatus	Last_Error	string	MISSING
mysql	SlaveStatus	Last_IO_Errno	long	MISSING
mysql	SlaveStatus	Last_IO_Error	string	MISSING
mysql	SlaveStatus	Last_SQL_Errno	long	MISSING
mysql	SlaveStatus	Last_SQL_Error	string	MISSING
mysql	SlaveStatus	Master_Host	string	MISSING
mysql	SlaveStatus	Master_ip	string	MISSING
mysql	SlaveStatus	Master_Log_File	string	MISSING
mysql	SlaveStatus	Master_Port	long	MISSING

Namespace	Namespace Type	Attribute	Type	Description
mysql	SlaveStatus	Master_SSL_Allowed	string	MISSING
mysql	SlaveStatus	Master_SSL_CA_File	string	MISSING
mysql	SlaveStatus	Master_SSL_CA_Path	string	MISSING
mysql	SlaveStatus	Master_SSL_Cert	string	MISSING
mysql	SlaveStatus	Master_SSL_Cipher	string	MISSING
mysql	SlaveStatus	Master_SSL_Key	string	MISSING
mysql	SlaveStatus	Master_SSL_Verify_Server_Cert	string	MISSING
mysql	SlaveStatus	Master_User	string	MISSING
mysql	SlaveStatus	Master_uuid	string	MISSING
mysql	SlaveStatus	Position	long	MISSING
mysql	SlaveStatus	Read_Master_Log_Pos	long	MISSING
mysql	SlaveStatus	Relay_Log_File	string	MISSING
mysql	SlaveStatus	Relay_Log_Pos	long	MISSING
mysql	SlaveStatus	Relay_Log_Space	long	MISSING
mysql	SlaveStatus	Relay_Master_Log_File	string	MISSING
mysql	SlaveStatus	Replicate_Do_DB	string	MISSING
mysql	SlaveStatus	Replicate_Do_Table	string	MISSING
mysql	SlaveStatus	Replicate_Ignore_DB	string	MISSING
mysql	SlaveStatus	Replicate_Ignore_Table	string	MISSING
mysql	SlaveStatus	Replicate_Wild_Do_Table	string	MISSING
mysql	SlaveStatus	Replicate_Wild_Ignore_Table	string	MISSING
mysql	SlaveStatus	Seconds_Behind_Master	long	MISSING
mysql	SlaveStatus	Skip_Counter	long	MISSING
mysql	SlaveStatus	Slave_IO_Running	string	MISSING
mysql	SlaveStatus	Slave_IO_State	string	MISSING
mysql	SlaveStatus	Slave_SQL_Running	string	MISSING
mysql	SlaveStatus	Until_Condition	string	MISSING
mysql	SlaveStatus	Until_Log_File	string	MISSING
mysql	SlaveStatus	Until_Log_Pos	long	MISSING
mysql	sp_with_select_star	routine	string	MISSING
mysql	Statement	bytes	long	MISSING
mysql	Statement	comment	string	MISSING
mysql	Statement	connection_id	long	MISSING
mysql	Statement	database	string	MISSING
mysql	Statement	errors	long	MISSING
mysql	Statement	exec_time	long	MISSING
mysql	Statement	explain_plan	string	MISSING
mysql	Statement	host_from	string	MISSING
mysql	Statement	host_to	string	MISSING
mysql	Statement	no_good_index_used	long	MISSING
mysql	Statement	no_index_used	long	MISSING

Namespace	Namespace Type	Attribute	Type	Description
mysql	Statement	query_type	string	MISSING
mysql	Statement	rows	long	MISSING
mysql	Statement	source_location	string	MISSING
mysql	Statement	text	string	MISSING
mysql	Statement	user	string	MISSING
mysql	Statement	warnings	long	MISSING
mysql	StatementAnalysisSupport	auto_explain_min_exec_time_ms	long	MISSING
mysql	StatementAnalysisSupport	capture_examples	string	MISSING
mysql	StatementAnalysisSupport	capture_explain	string	MISSING
mysql	StatementAnalysisSupport	enabled	string	MISSING
mysql	StatementAnalysisSupport	frequency	string	MISSING
mysql	StatementSummary	avg_exec_time	string	MISSING
mysql	StatementSummary	bytes	long	MISSING
mysql	StatementSummary	bytes_rel	long	MISSING
mysql	StatementSummary	count	long	MISSING
mysql	StatementSummary	count_rel	long	MISSING
mysql	StatementSummary	database	string	MISSING
mysql	StatementSummary	errors	long	MISSING
mysql	StatementSummary	exec_time	long	MISSING
mysql	StatementSummary	exec_time_rel	long	MISSING
mysql	StatementSummary	max_bytes	long	MISSING
mysql	StatementSummary	max_exec_time	long	MISSING
mysql	StatementSummary	max_rows	long	MISSING
mysql	StatementSummary	min_bytes	long	MISSING
mysql	StatementSummary	min_exec_time	long	MISSING
mysql	StatementSummary	min_rows	long	MISSING
mysql	StatementSummary	no_good_index_used	long	MISSING
mysql	StatementSummary	no_index_used	long	MISSING
mysql	StatementSummary	query_type	string	MISSING
mysql	StatementSummary	rows	long	MISSING
mysql	StatementSummary	rows_rel	long	MISSING
mysql	StatementSummary	text	string	MISSING
mysql	StatementSummary	text_hash	string	MISSING
mysql	StatementSummary	warnings	long	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
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
mysql	status	Com_alter_db	long_counter	Count of ALTER DATABASE statements
mysql	status	Com_alter_db_upgrade	long_counter	MISSING
mysql	status	Com_alter_event	long_counter	MISSING
mysql	status	Com_alter_function	long_counter	MISSING
mysql	status	Com_alter_procedure	long_counter	MISSING
mysql	status	Com_alter_server	long_counter	MISSING
mysql	status	Com_alter_table	long_counter	Count of ALTER TABLE statements
mysql	status	Com_alter_tablespace	long_counter	MISSING
mysql	status	Com_analyze	long_counter	Count of ANALYZE statements
mysql	status	Com_assign_to_keycache	long_counter	MISSING
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	MISSING
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	MISSING
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	MISSING
mysql	status	Com_create_server	long_counter	MISSING
mysql	status	Com_create_table	long_counter	Count of CREATE TABLE statements
mysql	status	Com_create_trigger	long_counter	MISSING
mysql	status	Com_create_udf	long_counter	MISSING
mysql	status	Com_create_user	long_counter	Count of CREATE USER statements
mysql	status	Com_create_view	long_counter	MISSING
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	MISSING

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_drop_function	long_counter	Count of DROP FUNCTION statements
mysql	status	Com_drop_index	long_counter	Count of DROP INDEX statements
mysql	status	Com_drop_procedure	long_counter	MISSING
mysql	status	Com_drop_server	long_counter	MISSING
mysql	status	Com_drop_table	long_counter	Count of DROP TABLE statements
mysql	status	Com_drop_trigger	long_counter	MISSING
mysql	status	Com_drop_user	long_counter	Count of DROP USER statements
mysql	status	Com_drop_view	long_counter	MISSING
mysql	status	Com_empty_query	long_counter	MISSING
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	MISSING
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	MISSING
mysql	status	Com_rename_table	long_counter	Count of RENAME TABLE statements
mysql	status	Com_rename_user	long_counter	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_rollback	long_counter	Count of ROLLBACK statements
mysql	status	Com_rollback_to_savepoint	long_counter	MISSING
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	MISSING
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	MISSING
mysql	status	Com_show_create_db	long_counter	Count of SHOW CREATE DATABASE statements
mysql	status	Com_show_create_event	long_counter	MISSING
mysql	status	Com_show_create_func	long_counter	MISSING
mysql	status	Com_show_create_proc	long_counter	MISSING
mysql	status	Com_show_create_table	long_counter	Count of SHOW CREATE TABLE statements
mysql	status	Com_show_create_trigger	long_counter	MISSING
mysql	status	Com_show_databases	long_counter	Count of SHOW DATABASES statements
mysql	status	Com_show_engine_logs	long_counter	MISSING
mysql	status	Com_show_engine_mutex	long_counter	MISSING
mysql	status	Com_show_engine_status	long_counter	MISSING
mysql	status	Com_show_errors	long_counter	Count of SHOW ERRORS statements
mysql	status	Com_show_events	long_counter	MISSING
mysql	status	Com_show_fields	long_counter	Count of SHOW FIELDS statements
mysql	status	Com_show_function_status	long_counter	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
				statements
mysql	status	Com_show_plugins	long_counter	MISSING
mysql	status	Com_show_privileges	long_counter	Count of SHOW PRIVILEGES statements
mysql	status	Com_show_procedure_status	long_counter	MISSING
mysql	status	Com_show_processlist	long_counter	Count of SHOW PROCESSLIST statements
mysql	status	Com_show_profile	long_counter	MISSING
mysql	status	Com_show_profiles	long_counter	MISSING
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	MISSING
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	MISSING
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	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
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
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

Namespace	Namespace Type	Attribute	Type	Description
				in a table
mysql	status	Innodb_buffer_pool_pages_data	long	The number of pages containing data (dirty or clean)
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

Namespace	Namespace Type	Attribute	Type	Description
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 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

Namespace	Namespace Type	Attribute	Type	Description
				block from the cache
mysql	status	Key_reads	long_counter	The number of physical reads of a key block from disk
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	Ndb_number_of_data_nodes	long	MISSING
mysql	status	Ndb_number_of_ready_data_nodes	long	MISSING
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	MISSING
mysql	status	Open_tables	long	The number of tables that are open
mysql	status	Opened_files	long_counter	MISSING
mysql	status	Opened_table_definitions	long_counter	MISSING
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	Queries	long_counter	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
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 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_tim	long_counter	The number of session cache timeouts

Namespace	Namespace Type	Attribute	Type	Description
		eouts		
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
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	status	Uptime_since_flush_status	long_counter	MISSING
mysql	strong_privileges	user	string	MISSING
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	tables_no_myisam_index_stats	table_list	string	MISSING

Namespace	Namespace Type	Attribute	Type	Description
mysql	tables_no_unique_keys	table_list	string	MISSING
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
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	MISSING
mysql	transactions_in_serial_log	transactions_in_serial_log	long	MISSING
mysql	trigger_with_select_star	trigger_defn	string	MISSING
mysql	user_on_missing_db	db_name	string	MISSING
mysql	user_on_missing_db	user	string	MISSING
mysql	user_on_missing_table	user	string	MISSING
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	MISSING
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	MISSING
mysql	variables	binlog_cache_size	long	Memory.Caches
mysql	variables	binlog_direct_non_transactional_updates	string	MISSING
mysql	variables	binlog_format	string	MISSING

Namespace	Namespace Type	Attribute	Type	Description
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
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	MISSING
mysql	variables	event_scheduler	string	MISSING
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	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
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	MISSING
mysql	variables	general_log_file	string	MISSING
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	MISSING
mysql	variables	have_compress	string	Availability of the zlib compression library
mysql	variables	have_crypt	string	Availability of the crypt() system call
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	MISSING
mysql	variables	have_profiling	string	MISSING
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	MISSING
mysql	variables	ignore_builtin_innodb	string	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
				startup
mysql	variables	init_slave	string	Command(s) that are executed when a slave connects to a master
mysql	variables	innodb_adaptive_flushing	string	MISSING
mysql	variables	innodb_adaptive_hash_index	string	MISSING
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	MISSING
mysql	variables	innodb_buffer_pool_awe_mem_mb	long	Memory.Buffers
mysql	variables	innodb_buffer_pool_size	long	Memory.Buffers
mysql	variables	innodb_change_buffering	string	MISSING
mysql	variables	innodb_checksums	string	Enable InnoDB checksums validation
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_format	string	MISSING
mysql	variables	innodb_file_format_check	string	MISSING
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_io_capacity	long	MISSING
mysql	variables	in-	long	Table Types.InnoDB

Namespace	Namespace Type	Attribute	Type	Description
		nodb_lock_wait_timeout		
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_old_blocks_pct	long	MISSING
mysql	variables	innodb_old_blocks_time	long	MISSING
mysql	variables	innodb_open_files	long	The maximum number of files that InnoDB keeps open at the same time
mysql	variables	innodb_read_ahead_threshold	long	MISSING
mysql	variables	innodb_read_io_threads	long	MISSING
mysql	variables	innodb_rollback_on_timeout	string	Unknown.Unknown
mysql	variables	innodb_spin_wait_delay	long	MISSING
mysql	variables	innodb_stats_on_metadata	string	MISSING
mysql	variables	innodb_stats_sample_pages	long	MISSING
mysql	variables	innodb_strict_mode	string	MISSING
mysql	variables	innodb_support_xa	string	Enable InnoDB support for the XA 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

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	innodb_thread_concurrency	long	Table Types.InnoDB
mysql	variables	innodb_thread_sleep_delay	long	Table Types.InnoDB
mysql	variables	innodb_use_legacy_cardinality_algorithm	string	MISSING
mysql	variables	innodb_use_sys_malloc	string	MISSING
mysql	variables	innodb_write_io_threads	long	MISSING
mysql	variables	insert_id	long	MISSING
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	MISSING
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	MISSING
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	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
				log file
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	MISSING
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_mmap_size	long	MISSING
mysql	variables	myis-	string	The value of the --myisam-recover op-

Namespace	Namespace Type	Attribute	Type	Description
		am_recover_options		tion
mysql	variables	myisam_repair_threads	long	Table Types.MyISAM
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	MISSING
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	MISSING
mysql	variables	old_alter_table	string	MISSING
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	optimizer_switch	string	MISSING
mysql	variables	pid_file	string	Pid file used by safe_mysqld
mysql	variables	plugin_dir	string	MISSING
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	profiling	string	MISSING
mysql	variables	profiling_history_size	string	MISSING
mysql	variables	protocol_version	long	The version of the client/server protocol used by the MySQL server
mysql	variables	pseudo_thread_id	long	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	query_prealloc_size	long	Persistent buffer for query parsing and execution
mysql	variables	rand_seed1	long	MISSING
mysql	variables	rand_seed2	long	MISSING
mysql	variables	range_alloc_block_size	long	Allocation block size for storing ranges during optimization
mysql	variables	read_buffer_size	long	Memory.Buffers
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	MISSING
mysql	variables	relay_log_index	string	MISSING
mysql	variables	relay_log_info_file	string	MISSING
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	MISSING
mysql	variables	report_password	string	MISSING
mysql	variables	report_port	long	MISSING
mysql	variables	report_user	string	MISSING
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_name_resolve	string	MISSING
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	MISSING
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	MISSING
mysql	variables	slow_query_log_file	string	MISSING
mysql	variables	socket	string	Socket file to use for connection
mysql	variables	sort_buffer_size	long	Each thread that needs to do a sort al-

Namespace	Namespace Type	Attribute	Type	Description
				locates a buffer of this size
mysql	variables	sql_auto_is_null	string	MISSING
mysql	variables	sql_big_selects	string	General.Features
mysql	variables	sql_big_tables	string	MISSING
mysql	variables	sql_buffer_result	string	MISSING
mysql	variables	sql_log_bin	string	MISSING
mysql	variables	sql_log_off	string	MISSING
mysql	variables	sql_log_update	string	MISSING
mysql	variables	sql_low_priority_updates	string	MISSING
mysql	variables	sql_max_join_size	long	MISSING
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	MISSING
mysql	variables	sql_safe_updates	string	MISSING
mysql	variables	sql_select_limit	long	MISSING
mysql	variables	sql_slave_skip_counter	long	MISSING
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	MISSING
mysql	variables	table_lock_wait_timeout	long	General.Miscellaneous
mysql	variables	table_open_cache	long	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	thread_handling	string	MISSING
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	MISSING
mysql	variables	tmp_table_size	long	Memory.Overview
mysql	variables	tmpdir	string	Path for temporary files
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	MISSING
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	MISSING
mysql	waiting_db	waiting_db	string	MISSING
mysql	waiting_statement	waiting_statement	string	MISSING
mysql	waiting_thread	waiting_thread	long	MISSING
mysql	waiting_time	waiting_time	long	MISSING
mysql	waiting_user	waiting_user	string	MISSING
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	MISSING
os	disk	disk_servicetime	double	MISSING
os	disk	disk_snaptime	long	MISSING
os	disk	disk_time	long	MISSING
os	disk	disk_writes	long_counter	The number of disk writes

Namespace	Namespace Type	Attribute	Type	Description
os	disk	disk_wtime	long	MISSING
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 mount point
os	disk	fs_used	long	The amount of used space for the mount point
os	fs	fs_avail	long	MISSING
os	fs	fs_dev_name	string	MISSING
os	fs	fs_dir_name	string	MISSING
os	fs	fs_files	long	MISSING
os	fs	fs_flags	long	MISSING
os	fs	fs_free_files	long	MISSING
os	fs	fs_options	string	MISSING
os	fs	fs_sys_type_name	string	MISSING
os	fs	fs_total	long	MISSING
os	fs	fs_type_name	string	MISSING
os	fs	fs_unused	long	MISSING
os	fs	fs_used	long	MISSING
os	Host	host_id	string	The host unique identifier
os	Host	name	string	MISSING
os	loadavg	0	double	MISSING
os	loadavg	1	double	MISSING
os	loadavg	2	double	MISSING
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	MISSING
os	mem	swap_page_out	long	MISSING
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

Namespace	Namespace Type	Attribute	Type	Description
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
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	MISSING
util	LogHistogram	1	long_counter	MISSING
util	LogHistogram	10	long_counter	MISSING
util	LogHistogram	11	long_counter	MISSING
util	LogHistogram	12	long_counter	MISSING
util	LogHistogram	13	long_counter	MISSING
util	LogHistogram	14	long_counter	MISSING
util	LogHistogram	15	long_counter	MISSING
util	LogHistogram	16	long_counter	MISSING
util	LogHistogram	17	long_counter	MISSING

Namespace	Namespace Type	Attribute	Type	Description
util	LogHistogram	18	long_counter	MISSING
util	LogHistogram	19	long_counter	MISSING
util	LogHistogram	2	long_counter	MISSING
util	LogHistogram	20	long_counter	MISSING
util	LogHistogram	21	long_counter	MISSING
util	LogHistogram	22	long_counter	MISSING
util	LogHistogram	23	long_counter	MISSING
util	LogHistogram	24	long_counter	MISSING
util	LogHistogram	25	long_counter	MISSING
util	LogHistogram	26	long_counter	MISSING
util	LogHistogram	27	long_counter	MISSING
util	LogHistogram	28	long_counter	MISSING
util	LogHistogram	29	long_counter	MISSING
util	LogHistogram	3	long_counter	MISSING
util	LogHistogram	30	long_counter	MISSING
util	LogHistogram	31	long_counter	MISSING
util	LogHistogram	32	long_counter	MISSING
util	LogHistogram	33	long_counter	MISSING
util	LogHistogram	34	long_counter	MISSING
util	LogHistogram	35	long_counter	MISSING
util	LogHistogram	36	long_counter	MISSING
util	LogHistogram	37	long_counter	MISSING
util	LogHistogram	38	long_counter	MISSING
util	LogHistogram	39	long_counter	MISSING
util	LogHistogram	4	long_counter	MISSING
util	LogHistogram	40	long_counter	MISSING
util	LogHistogram	41	long_counter	MISSING
util	LogHistogram	42	long_counter	MISSING
util	LogHistogram	43	long_counter	MISSING
util	LogHistogram	44	long_counter	MISSING
util	LogHistogram	45	long_counter	MISSING
util	LogHistogram	46	long_counter	MISSING
util	LogHistogram	47	long_counter	MISSING
util	LogHistogram	48	long_counter	MISSING
util	LogHistogram	49	long_counter	MISSING
util	LogHistogram	5	long_counter	MISSING
util	LogHistogram	50	long_counter	MISSING
util	LogHistogram	51	long_counter	MISSING
util	LogHistogram	52	long_counter	MISSING
util	LogHistogram	53	long_counter	MISSING
util	LogHistogram	54	long_counter	MISSING
util	LogHistogram	55	long_counter	MISSING
util	LogHistogram	56	long_counter	MISSING
util	LogHistogram	57	long_counter	MISSING
util	LogHistogram	58	long_counter	MISSING

Namespace	Namespace Type	Attribute	Type	Description
util	LogHistogram	59	long_counter	MISSING
util	LogHistogram	6	long_counter	MISSING
util	LogHistogram	60	long_counter	MISSING
util	LogHistogram	61	long_counter	MISSING
util	LogHistogram	62	long_counter	MISSING
util	LogHistogram	7	long_counter	MISSING
util	LogHistogram	8	long_counter	MISSING
util	LogHistogram	9	long_counter	MISSING
util	LogHistogram	base	long	MISSING
util	LogHistogram	nul	long_counter	MISSING

Appendix E. Error codes

The following table documents error codes reported by the GUI and the associated message.

Table E.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.
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.
U0130	The uploaded Graph had an insufficient version number, and was not loaded.

Error Code	Message
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.
U0165	Heat chart rule "*replaceable_value*" scheduled against "*replaceable_value*" may not be unsched-

Error Code	Message
	uled. 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.
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.
U0208	Nothing is selected.

Error Code	Message
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 user search criteria.
U0213	You must choose user search by pattern or by criteria.
U0214	The uploaded Advisor jar is a Custom Rule/Graph Export Bundle, and must be uploaded to the Import/Export page.

The following table documents error codes reported by the Enterprise server and the associated message.

Table E.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.
E0304	Group element "*replaceable_value*" already exists.
E0305	Group element "*replaceable_value*" does not exist in the group "*replaceable_value*".

Error Code	Message
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.
E0904	Invalid server list.
E0905	SKIPPED "*replaceable_value*" for *replaceable_value* because Agent for *replaceable_value* is down.

Error Code	Message
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.
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.

Error Code	**Message**
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 F. 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

- F.1: My MySQL Enterprise Service Manager is behind a firewall but it cannot communicate with the MySQL Enterprise website. I normally use a proxy service to access external websites. How do I configure the proxy settings for MySQL Enterprise Dashboard?
- F.2: How do I change the name of a server?
- F.3: 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?
- F.4: 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.
- F.5: 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.
- F.6: I have enabled `EXPLAIN` queries for Query Analyzer, but no queries with the `EXPLAIN` data are showing up in the display.
- F.7: Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?
- F.8: Why do some rules appear to have a **Severity** of **Unknown**?
- F.9: What is the relationship between the advisor JAR file and the key?
- F.10: Does the Gold-level key support Silver-level advisors?
- F.11: Can the Trial-level key work with the Gold-level advisors JAR file?
- F.12: Can I run MySQL Enterprise Service Manager on machine with other applications running?
- F.13: How frequently is the data purge process executed?

Questions and Answers

F.1: My MySQL Enterprise Service Manager is behind a firewall but it cannot communicate with the MySQL Enterprise website. 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, edit the `apache-tomcat/conf/catalina.properties` file within the MySQL Enterprise Service Manager installation directory. Change the `http.proxyHost` and `http.proxyPort` properties:

```
http.proxyHost=proxy.example.com
http.proxyPort=8080
```

Restart the MySQL Enterprise Service Manager for the change to take effect:

```
shell> mysqlmonitorctl.sh restart
```

F.2: 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.

F.3: 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.

F.4: 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 are updated according to their refresh period, regardless of the refresh period set for the main display page.

F.5: 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 3.2.3, “Query Analyzer Settings”.

F.6: 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 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 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, modify this line:

```
auto_explain_min_exec_time_us = 100 * 1000
```

These changes take effect without restarting MySQL Enterprise Agent.

F.7: 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 Agent are using MySQL 5.0 or later. Clients that use the library provided with MySQL 4.1 or earlier do not work with MySQL Enterprise Agent.

F.8: 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 that is expected to be resolved in future versions of MySQL Enterprise Monitor.

F.9: 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.

F.10: 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.

F.11: 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.

F.12: Can I run MySQL Enterprise Service Manager on machine with other applications running?

You can, but Oracle recommends running your MySQL Enterprise Service Manager on a dedicated machine, especially when monitoring many agents.

F.13: 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

- F.1: What is MySQL Enterprise?
- F.2: Can I buy MySQL Enterprise subscriptions for multiple years?
- F.3: Can I buy MySQL Enterprise subscriptions for only some of my production MySQL database servers?
- F.4: Do all my MySQL Enterprise subscriptions need to be at the same tier?
- F.5: What if I plan to add more MySQL servers to my MySQL Enterprise subscription?
- F.6: How should I decide between MySQL Enterprise Basic, Silver, Gold and Platinum?
- F.7: What is MySQL Enterprise Server?
- F.8: What is MySQL Production Support?
- F.9: Does MySQL Enterprise include 24x7 Technical Support?
- F.10: Does MySQL Enterprise include Maintenance, Updates, and Upgrades?
- F.11: Does MySQL Enterprise include Emergency Hot Fix Builds?
- F.12: What is MySQL Consultative Support?
- F.13: What is a Technical Account Manager?
- F.14: Does MySQL provide IP (Intellectual Property) Indemnification?
- F.15: What is the list of Supported Platforms?
- F.16: Are there any Demo/Tutorials available for MySQL Enterprise?
- F.17: Are there any MySQL Enterprise White Papers available?

- F.18: Are there any Webinars available?
- F.19: What is the pricing of MySQL Enterprise?
- F.20: What is MySQL Enterprise Unlimited?
- F.21: How do I get a 30-day trial on MySQL Enterprise?
- F.22: How do I buy MySQL Enterprise?

Questions and Answers

F.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

F.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

F.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

F.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.

F.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

F.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

F.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

F.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

F.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

F.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.

F.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.

F.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

F.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

F.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

F.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/

F.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

F.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

F.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

F.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

F.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

F.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

F.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

- F.1: What is MySQL Enterprise Monitor?
- F.2: What MySQL Enterprise subscription levels include the MySQL Enterprise Monitor?
- F.3: What are the features and related benefits of the MySQL Enterprise Monitor?
- F.4: What are the immediate benefits of implementing the MySQL Enterprise Monitor?
- F.5: What are the long-term benefits of the MySQL Enterprise Monitor?
- F.6: How is the MySQL Enterprise Monitor installed and deployed?
- F.7: How is the Enterprise Monitor web application architected?
- F.8: What makes MySQL Enterprise unique?
- F.9: What versions of MySQL are supported by the MySQL Enterprise Monitor?
- F.10: What operating system platforms are supported by the MySQL Enterprise Monitor?
- F.11: How do I get the MySQL Enterprise Monitor?
- F.12: What are the MySQL Enterprise Advisors and Advisor Rules?
- F.13: Which Advisors and features are included under different MySQL Enterprise subscription levels?
- F.14: Which set of Enterprise Advisors, Advisor Rules and features are best for my use of MySQL?
- F.15: How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?

Questions and Answers

F.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

F.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

F.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.

F.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

F.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.

F.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.

F.7: How is the Enterprise Monitor web application architected?

The Enterprise Monitor web application comprises three 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 via 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.

F.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.

F.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.

F.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.

F.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

F.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 comprise 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 additional resources for additional research into the issue at hand. http://www.mysql.com/products/enterprise/advisors.html

F.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.

F.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.

F.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 via 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

- F.1: What is the MySQL Query Analyzer?
- F.2: How is the MySQL Query Analyzer installed and enabled?
- F.3: What overhead can I expect when the MySQL Query Analyzer is installed and enabled?
- F.4: Can I leave the MySQL Query Analyzer enabled at all times?
- F.5: What are the main features and benefits of the MySQL Query Analyzer?
- F.6: What are the typical use cases of the MySQL Query Analyzer?
- F.7: How are subscribers notified about updates to the MySQL Query Analyzer application components?
- F.8: What makes the MySQL Query Analyzer unique?
- F.9: How can I get the MySQL Query Analyzer?
- F.10: Does Query Analyzer work with MySQL Cluster?
- F.11: Does Query Analyzer capture queries by the root user?
- F.12: Does Query Analyzer enable me to monitor the disk reads and writes during a query?
- F.13: Does Query Analyzer handler prepared statements?
- F.14: How much degradation in performance does mysql-proxy introduce?
- F.15: Does the query analyzer look at all queries? or only queries which would show up in the in the slow-queries log?
- F.16: Does the "Rows" area show the rows returned/updated or the rows visited by the query?
- F.17: Do the MySQL clients have to connect to the port of the mysql proxy to enable the QA?

Questions and Answers

F.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

F.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.

F.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.

F.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.

F.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

F.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.

F.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 via 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.

F.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.

F.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

F.10: Does Query Analyzer work with MySQL Cluster?

Yes, providing that exact node is monitored with an agent and query analyzer has been enabled for that node. Note that you must be accessing your cluster data through a standard MySQL node for this to work.

F.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 Agent.

F.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.

F.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.

F.14: How much degradation in performance does mysql-proxy introduce?

At the very least it's equivalent to a network hop in latency. The degradation 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 noticeable. 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.

F.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.

F.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.

F.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 G. 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.

G.1. Changes in MySQL Enterprise Monitor 2.2.3 (8 September 2010)

Bugs fixed:

- When performing an unattended installation, the MySQL Enterprise Service Manager would not be restarted automatically if the configuration file contained changes. You can now force the restart to occur by using the `--forceRestart` option. When set to 1, the server will be restarted using the old (unmodified) configuration file. (Bug#56192)
- The content of the user field on the **Edit User** screen was not protected so it was possible to insert scripting instructions into the field contents. (Bug#56189)
- It was possible to fake the request and promote a user to a different role without having the privileges to do so. (Bug#56187)
- The `--skip-locking` option for the embedded MySQL server was being used in place of the deprecated `--skip-external-locking` option. (Bug#56004)
- The embedded OpenSSL has been updated to version 0.9.8o. (Bug#55949)
- It was possible for a user without the ability to view the **Manage Users** page by visiting the URL, even if the user was not authorized to view the page. Users without the right credentials are now provided with a warning that the page is unavailable. (Bug#55585)
- When the MySQL server supporting MySQL Enterprise Service Manager is unavailable, the `catalina.out` file is populated with error messages for each MySQL Enterprise Agent communicating with MySQL Enterprise Service Manager. The quantity and frequency of the messages can cause the error log to grow at a very fast rate. The logging configuration has been altered so that Tomcat-specific errors go to `tomcat.log`, rotating every 10MB for up to 10 logs. Application errors are reported in `mysql-monitor.log`. (Bug#55516)
- The same example query could be consistently resent to MySQL Enterprise Service Manager, even though the example query had already been reported. (Bug#55453)
- Using LDAP authentication for mapping roles using user attributes would not authenticate correctly. (Bug#55450)
- The supplied `setenv.sh` script sets the wrong temporary directory value for `HeapDumpPath`. The path has been updated to the `monitor/apache-tomcat/temp` directory within the MySQL Enterprise Service Manager installation. (Bug#54533)
- The MySQL Enterprise Agent installer would ask for agent credentials, even if you had indicated that the correct user credentials had already been created. (Bug#53943)
- The Tomcat `config.properties` file could be installed with unsafe permissions. (Bug#52985)
- The file permissions on the `mysql` directory within the MySQL Enterprise Service Manager installation would not be set correctly on Mac OS X. (Bug#35203)

G.2. Changes in MySQL Enterprise Monitor 2.2.2 (15 July 2010)

Bugs fixed:

- LDAP roles using the search method did not work when searching for authentication credentials. (Bug#54806)
- The output from MySQL Enterprise Agent using `--version` would show incorrect versions for some components. (Bug#54701)
- The data reported for the binary log file size (in `mysql::masterlogs::filesizesum`) was erroneously reporting the count of the files, not the sum of the size of the binary logs. (Bug#54618)

- Monitoring a database with a large number of tables would lead to long queries on INFORMATION_SCHEMA which could lead to operating system memory graphs not being updated correctly. (Bug#54591)
- The help output from MySQL Enterprise Agent could show missing or corrupt characters. (Bug#54559)
- The MySQL Enterprise Agent could fail to start on 64-bit platforms. (Bug#54433)
- Using the diagnostic zip functionality would fail if you had configured a server group with a forward slash / in the name. (Bug#54136)
- Using the Error option for the Referral method for LDAP settings generates an error trace. (Bug#54047)
- A MySQL Enterprise Agent could report the a MySQL instance as being down, even though the instance was available. This was related to the way local and remote filesystems were identified and reported by MySQL Enterprise Agent. A new option, --agent-report-network-fs has been added to indicate whether remote filesystems should be reported in the diskspace reports. By default, the MySQL Enterprise Agent no longer reports network filesystem information, only data for local filesystems. (Bug#54002)
- The ssh fingerprint used to identify agents would fail on Ubuntu Linux. (Bug#53977)
- On Windows platforms, the MySQL Enterprise Agent could show increasing memory usage over time. (Bug#53880)
- When mysqld is in a hung state and unresponsive, MySQL Enterprise Agent and MySQL Enterprise Service Manager may not report the server as unavailable. (Bug#53090)
- If a MySQL Enterprise Agent got a timeout during the initial checkin with MySQL Enterprise Service Manager (for instance, if MySQL Enterprise Service Manager was busy), it would fail to resynchronize properly and show the monitored MySQL instances as down. (Bug#52952)
- The GRANT for the agent user could include permissions for a wide range ofd hosts, and different grant sould be created by different platform installers. (Bug#52933)
- During and upgrade installation from 2.1 to 2.2, some files in the MySQL Enterprise Agent installation directory would remain. During an upgrade, the installer now performs the following operations:

  - The file custom.xml will be copied from share/mysql-proxy/items to share/mysql-monitor-agent/items.
  - The contents of share/mysql-proxy and the directory are deleted.
  - The README files in share/doc are removed. The 2.2 files are located in share/doc/mysql-monitor-agent.

  (Bug#52770)
- The installer dialog for the text installation for MySQL Enterprise Agent would show a blank password prompt when asking for confirmation of the root password. (Bug#52129)
- When performing an upgrade, the value of agent-host-id would not be preserved. (Bug#47676)
- A TRUNCATE TABLE statement could be mis-identified in the tokenizer. (Bug#44772)
- When reporting errors related to multiple instances monitored by the same MySQL Enterprise Agent, the error message would not indicate which instance reported the error. (Bug#44748, Bug#47374)
- Errors determining InnoDB data within MySQL Enterprise Agent would be reported, even when InnoDB had been disabled on the instance. (Bug#41168)

G.3. Changes in MySQL Enterprise Monitor 2.2.1 (26 May 2010)

Functionality added or changed:

- The Query Analyzer now includes a Transactional filter to cover all transaction related statements, in addition to the individual statement filters for BEGIN, START, COMMIT, ROLLBACK, SAVEPOINT, and RELEASE. (Bug#53564)

Bugs fixed:

- The required version for running a separate MySQL server has been updated to 5.1.43 in the MySQL Enterprise Service Manager installer. (Bug#53953)
- Query Analyzer could fail to operate correctly when upgrading from MySQL Enterprise Service Manager 2.1 to MySQL Enterprise Service Manager 2.2. Accessing the details of a query could lead to a `NullPointerException` error. (Bug#53845)
- It was possible to import a custom advisor bundle to replace the main advisor bundle. This would delete all existing rules, schedules and advisors. The system now checks the bundle before import. (Bug#53826)
- The default settings for the LDAP interface for authenticating users has been updated. By default, lookups are made using `Bind as user`, and the default password digest mechanism is now MD5. (Bug#53679)
- When generating a support report, the server provided no feedback on the production process. A page will now be generated to advise you of the current status before providing you with a completed message once the report generation has been completed. (Bug#53584)
- If the backend MySQL database being used for MySQL Enterprise Service Manager is unavailable during an update installation, the installer may give a misleading message indicating the problem. (Bug#53557)
- The filters for `LOAD DATA INFILE`, `START TRANSACTION`, and `HANDLER CLOSE` statements in the Query Analyzer interface did not filter statements correctly. (Bug#53425)
- A `LOAD DATA INFILE` statement would not get normalized correctly within the Query Analyzer display. (Bug#53419)
- If there are directories located within the configuration instances directory that are empty (i.e. contain no instance configuration file), MySQL Enterprise Agent would fail to start. The behavior has been updated to provide a `message` level error. (Bug#53317)
- A warning is now provided before generating the support diagnostics information to indicate the potential time delay. (Bug#53193)
- When upgrading an existing MySQL Enterprise Service Manager installation, where the MySQL database is identified as not being on the current host (either because it is on the same machine with a different IP, or a different machine), the upgrade installer would fail to provide a warning that such a MySQL database installation cannot be upgraded automatically by the installer. A warning providing more information on how to upgrade is now provided. (Bug#53158)
- Adding a new instance to the list of monitored instances while using the MySQL Enterprise Dashboard could lead to a scheduling exception error. (Bug#52190)
- Deleting a user while the user is currently logged in and using MySQL Enterprise Dashboard would not prevent the user from continuing to use the MySQL Enterprise Dashboard, even though the user no longer existed in the system. (Bug#52142)
- The methods for replication discovery by MySQL Enterprise Agent have been modified. A new option for MySQL Enterprise Agent has been added that prevents the master UUID being used in the replication topology determination. This also prevents MySQL Enterprise Agent from obtaining replication information from the `master.info` file. (Bug#50604)
- If the backend MySQL database being used for MySQL Enterprise Service Manager is unavailable during an update installation, the installer may give a misleading message indicating the problem. (Bug#38897)
- If a Heat Chart rule had de-escalated and the corresponding higher event had not been closed, the Heat Chart dot link did not show the event on the Events tab. This was because the de-escalated state was shown in the Heat Chart, but an Event was not generated for it until the more serious Event was closed. (Bug#28529)

G.4. Changes in MySQL Enterprise Monitor 2.2.0 (5 May 2010)

Functionality added or changed:

- The included Java runtime environment has been updated to JDK 6.0 Update 18. (Bug#50347)
- You can delete multiple servers from the Manage Servers tab by selecting each server and using the DELETE SERVERS button. (Bug#50259)

- The MySQL Enterprise Dashboard has been confirmed as compatible with Google Chrome 4.0. (Bug#49407)
- The **Manage Servers** window has been updated to allow you to view and filter the list of servers using a variety of criteria, making it easier to find servers, agents, or both that require upgrading. (Bug#48329)
- The version of MySQL Server installed with MySQL Enterprise Monitor was upgraded to 5.1.40. (Bug#48218)
- The smallest purge log time interval that could be set in the Dashboard was one week. MySQL Enterprise Monitor was changed to allow setting the smallest purge log time interval to one day. (Bug#46822)
- Certain heat chart rules cannot be unscheduled or disabled, this is because they are required for correct operation of MySQL Enterprise Monitor. Should an attempt be made to unschedule or disable one of these heat chart rules, a suitable message is now displayed, explaining this requirement. (Bug#46535)
- Custom graphs and advisors can now be imported, exported and deleted from the system. See Section 2.8.8, “Import/Export”. (Bug#45436)
- The **Manage Servers** page was changed so that it was possible to view the Up Time for a server by hovering the mouse cursor over the value in the **Up Since** column. Additionally, the ability to sort servers based on the **Up Since** column was also added. (Bug#44309)
- The **Manage Servers** page was changed to include the Agent version number, and additionally the ability to sort servers based on the Agent version column. (Bug#44308)
- Support for LDAP authentication has been added. (Bug#43031)
- The Query Analyzer interface has been improved to show more information about query states and for the filtering system to enable you to view and filter the contents to more easily identify problem queries. For more information, see Section 3.2.4, “Filtering Query Analyzer Data”. (Bug#41643)
- When running an update installation on a machine where you have multiple instances of MySQL Enterprise Agent already installed, the installer could identify and upgrade the wrong version. You can now specify the installed version (as identified by the startup script in `/etc/init.d`) by using the `--servicename=`*`mysql-monitor-agent`* option to the command line installer.

  For new installations where a instance of the MySQL Enterprise Agent is already installed, the installer could identify an existing version and refuse to perform the installation process. During the installation, you are now prompted with the installation name to use for the startup script if, and only if, an existing instance has been identified. (Bug#41005)
- 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 disallowed special characters can be found at the following location: http://en.wikipedia.org/wiki/Percent-encoding#Types_of_URI_characters (Bug#37172)

Bugs fixed:

- The full agent URL would be output in the logs in the event of an error, including the agent username and password used to connect to MySQL Enterprise Service Manager. The password is now stripped from the reported URL. (Bug#53199)
- During installation of MySQL Enterprise Service Manager on Windows, a batch file was created to obtain the MySQL version number. This file could contain security credentials for accessing the server. The batch file is no longer created. (Bug#52847)
- Uninstalling MySQL Enterprise Service Manager using the Uninstall Services tool, or later removing MySQL Enterprise Service Manager after servers have already been successfully removed, could fail. (Bug#52846)
- The filter selection in the drop down menus in the **Manage Servers** tab have been changed from blank entries (indicating all items), to `(all)`. (Bug#52800)
- `EXPLAIN` output within the Query Analyzer data would not be rendered correctly if some of the returned `EXPLAIN` plan data contained `NULL` values. (Bug#52532)
- When installing MySQL Enterprise Service Manager, the incorrect permissions could be applied to the embedded MySQL server. (Bug#52213)
- When performing an installation of MySQL Enterprise Service Manager or MySQL Enterprise Agent and specifying an alternative

service name during installation, the installer would recommend using the generic, not custom, service management script to initiate the service. (Bug#52198)

- Creating a new custom rule without a configured variable would not raise an error. The system will now return `U0068 One or more variables is required to make a functioning rule.`. (Bug#51218)

- During an upgrade installation of MySQL Enterprise Service Manager, custom SSL certificates that had been installed could be replaced by the standard versions. Existing custom certificates are now preserved during an upgrade. (Bug#51118)

- Scheduling a rule against an instance, and then deleting that instance, would retain the instance within MySQL Enterprise Dashboard, identified as `Unknown`. (Bug#51095)

- If the MySQL server holding the repository information for MySQL Enterprise Service Manager crashes and you are using MySQL Proxy to monitor the backend server, the problem may be difficult to identify from the logs. (Bug#50881)

- When using the MySQL Enterprise Agent Proxy Service, if the backend MySQL server went down, and then the clock on the MySQL Enterprise Agent Proxy Service host went back in time (for example, during daylight savings time adjustments), the MySQL Enterprise Agent Proxy Service would stop sending queries to the configured backend. (Bug#50806)

- The SSL certificates supplied with MySQL Enterprise Service Manager have been updated.

  The certificate shipped with MySQL Enterprise Service Manager is an example certificate that expires after 1 year and that folks should create their own and back it up between MEM service manager updates. For more information on updating the certificate, see Section C.1.2, “Creating a new SSL KeyStore”. (Bug#50694)

- Clicking a link to a support issue within the dashboard is now configured to open the issue within a new window. (Bug#50651)

- To aid in the replication discovery performed by the agent, you can now provide alternative credentials for connecting to the replication master to obtain the master UUID created by MySQL Enterprise Monitor. The master UUID enables MySQL Enterprise Agent to more easily identify and reconcile the master. You set these options on an instance basis, and setting the options prevents the agent using the `master.info` to determine the master UUID. (Bug#50604)

- The MyISAM Key Buffer Usage Graph could report misleading information about the key buffer block usage allocation. (Bug#50365)

- When the agent is shut down, the message used to be reported at the `critical` log level. This has been changed so that shutdown message are now reported at the `message` log level. You can enforce the shutdown message to be reported in the log regardless of the log level by using the `verbose-shutdown` option. (Bug#50340)

- The installer for MySQL Enterprise Agent on Linux 64-bit using glibc-2.3 would fail before the installation had completed properly. (Bug#50289)

- Deleting users within MySQL Enterprise Service Manager could lead to errors in the repository database that would affect further operations involving the deleted user. (Bug#49896)

- When creating a new instance by copying an existing agent configuration, it is possible to create an orphaned agent. The recommended advice is to `TRUNCATE` the `mysql.inventory` table. However, doing this could lead to additional errors and an exception when the scheduled data updates on the now orphaned agent are executed. (Bug#49882)

- The `configuration_report.txt` file would cont6ain information about the SSL port to use, even if SSL support had not been enabled during installation. (Bug#49769)

- The memory footprint of MySQL Enterprise Agent would slowly increase over time, particularly on Mac OS X and Solaris/OpenSolaris. (Bug#49739)

- If the `agent-host-id` value contained a period, information would fail to be reported to MySQL Enterprise Service Manager. (Bug#49699)

- When starting up MySQL Enterprise Monitor, the relevant environment variables are now inserted into the `catalina.out` log duing startup.

  The environment variables reported include:

  ```
  $CATALINA_BASE
  $CATALINA_HOME
  $CATALINA_TMPDIR
  $JAVA_HOME
  ```

```
$JRE_HOME
$_RUNJAVA
$JAVA_OPTS
$CATALINA_OPTS
$JAVA_ENDORSED_DIRS
$CLASSPATH
```

In addition, the version information for the Java environment used will also be reported in the log. (Bug#49626)

- The information in the table on the **Query Analyzer** page was displayed in either ascending sort order, descending sort order, or an unsorted order. You could change this behavior by clicking the column you wanted to sort.

  The behavior has now been modified so that the table data is sorted in either ascending or descending order, clicking the columns toggles between these two states. (Bug#49471)

- The method for generating a new UUID for the MySQL Enterprise Agent has been changed. You can create a new UUID using:

  ```
  shell> bin/mysql-proxy --plugins=agent --agent-generate-uuid
  ```

  Alternatively, you can use the `bin/agent-generate-uuid` command in the agent installation directory. (Bug#49306, Bug#49903)

- When disabling or unscheduling Heat Chart rules, the following issues occurred:

  1. For the disable operation, the resulting page showed all rules as disabled, even though three rules were not disabled. However, if the page was refreshed, those three rules were correctly displayed as enabled.

  2. For the unschedule operation, there was no fading white cloud with rule counts.

  (Bug#48870)

- The backend Lua scripts used with the MySQL Proxy and MySQL Enterprise Agent were not updated for the the MySQL Proxy 0.8.0 core. This would lead to some Lua scripts triggering an error. (Bug#48718)

- The version information output for the MySQL Enterprise Agent did not include the version of the all the individual components loaded into the MySQL Enterprise Agent chassis. (Bug#48714)

- When using the proxy (for query analyzer) with the MySQL Enterprise Agent, the `proxy` plugin must be listed first in the `plugins` configuration parameter. (Bug#48713)

- The name of the section in `mysql-monitor-agent.ini` required for configuration is `mysql-proxy`, the default template had been mistakenly modified to `mysql-monitor-agent`. (Bug#48712)

- When upgrading an agent, the existing agent is shutdown and then restarted after the installation process has completed. To reduce downtime during installation, you can now opt to have the agent restarted immediately after installation, but before the rest of the installer completes. (Bug#48581)

- RAM statistics for FreeBSD machines would not be reported correctly by MySQL Enterprise Agent. (Bug#48493)

- Cutting and pasting sample queries from Query Analyzer page in could lead to bad concatenation of the query components. (Bug#48479, Bug#50121)

- When viewing pages of queries on the **Query Analyzer** tab, changing the filter options could lead to an empty list if you modified the filters while on a page that is higher than the number of the pages in the filtered display. The page numbering is now reset when you change the filter options to ensure that you are viewing the filtered queries. (Bug#48409)

- Changing the filter parameters on the **Query Analyzer** tab would lose your current sorting preferences. (Bug#48408)

- When viewing the **Query Analyzer** tab and selecting a server, the server name would still shown in the table of queries. When a server has been explicitly selected, the server column is no longer displayed. (Bug#48390)

- Deleting a server or group from the **Manage Servers** tab could result in a stacktrace dump if the server or group ID selected was no longer valid. A confirmation is now produced if MySQL Enterprise Service Manager identifies an issue when trying delete the selected server or group. (Bug#48255)

- When a time range was selected on a dashboard graph, the user was taken to BrowseQueries, and the correct graph and graph time range was selected. However, the following problems occurred:

  - The query time range did not agree with the graph time range.
  - Changes to the query range interval did not correctly update the graph or selection.
  - Changes to the graph type were not updated correctly.

  (Bug#48234)

- The URL for viewing the contents of an event alerted by email were not populated correctly, making the supplied URL invalid. (Bug#48193)
- The full-graph pop-up on the dashboard did not reflect the group/server selection of the containing dashboard, it always displayed the default group. (Bug#48167)
- The Add to Group context menu item, on the **Manage Servers** page, listed all servers when clicked, rather than the subset preselected from the main server list. (Bug#48141)
- When adding a new rule without providing a version a stacktrace would be produced. A default version (`1.0`) will be added to new rules if the version has not been specified. (Bug#47877)
- The indentation on alert emails could be incorrect if the list included a bullet list of items. (Bug#47802)
- When upgrading an installation of MySQL Enterprise Service Manager where the MySQL being used is not the bundled MySQL server, the upgrade would fail while creating a backup of the MySQL installation. (Bug#47775)
- Support has been added for SNMPv2 traps in addition to the existing SNMPv1 traps. You can configure the version to use for SNMP traps, see **SNMP Traps**.

  In addition, support has also been added to send SNMP notifications to two hosts in place of just one host. Notifications are sent to both hosts simultaneously. (Bug#47686, Bug#48955)
- Support issues can have a Severity from S1 through S4, but can also have a value of “NS” meaning “No Severity Set”. MySQL Enterprise Monitor was not able to parse this, which resulted in errors such as the following:

  ```
  2009-09-23 13:53:51,812 ERROR [em-worker-pool-thread-6:monitor.support.DevSpPoller] error
  consuming successful response
  java.lang.IllegalArgumentException: No enum const class
  com.mysql.etools.monitor.support.SpIssue$Severity.NS
          at java.lang.Enum.valueOf(Enum.java:196)
  ...
  ```

  (Bug#47562)
- MySQL Enterprise Monitor would fail to install on Miracle Linux4. (Bug#47209)
- Occasional Null Pointer Exception errors could be raised after an initial login to the dashboard due to problems determining the correct name to use for displaying the list of servers. (Bug#47181)
- The diagnostic report file has been updated so that the filename of the report includes the timestamp when the report was generated. (Bug#47164)
- The `REPLACE` and `CALL` statements did not show a rows graph within the statement pop-up graph tab. (Bug#46796)
- When trying to change the password for the agent user in the Dashboard to a password that contained a “@” or other special character, the following error message was displayed:

  ```
  An Error Occurred

      * U0160 The following characters are invalid in an agent password:

  !*'();:@&=+$,/?%#[]
  ```

  However, this password worked correctly when first connecting the agent user to the Monitor. (Bug#46757)

- The subject line in alert emails was not configurable. (Bug#46689)
- The schedule for the **MySQL Agent Not Reachable** rule cannot be modified. This is to ensure that problems communicating with the agent are identified immediately. If you attempt to modify the scheduling an error will be reported. (Bug#46667)
- Pressing cancel on a pop-up would cause a page reload, instead of just closing the pop-up window. (Bug#46604)
- Pressing cancel on a pop-up would cause a page reload, instead of just closing the pop-up window. (Bug#46604)
- The cry for help email included a stack trace, however it was displayed on a single line without any line breaks. (Bug#46458)
- Some queries would show multiple tables in the **Example Query** displayed within the Query Analyzer. (Bug#46438)
- On very large deployments with a large number of servers, the **Events** tab could take some time to load. (Bug#46436)
- When using MySQL Enterprise Agent, if the current configured time went backwards (for example during time correction), then the MySQL Enterprise Agent would stop reporting data and induce a high load on the machine running MySQL Enterprise Agent (Bug#46052)
- When viewing queries within the Query Analyzer, the example query shown could be taken from outside of the time range configured within the query analyzer view. (Bug#45683)
- The Windows installers for MySQL Enterprise Service Manager did not include 64-bit versions of the various binary tools. (Bug#45682)
- MySQL Enterprise Monitor alert emails used incorrect formatting.

  The following problems occurred:

  - Text in the subject and body was not formatted properly. Variables appeared to be displayed such as '.name', '.advice', and '.description', rather than values.
  - Line formatting (CR/LF) appeared to be absent.
  - No local time was shown (GMT time only).

  (Bug#44769)
- When a MySQL Enterprise Agent is disconnected from MySQL Enterprise Service Manager, the agent would keep 10 minutes of information by default, that would then be sent to the server when the agent reconnected. Any data older than 10 minutes would be lost. You can now configure the backlog timeout by using the `agent-backlog-timeout` to specify the amount of time that data will be kept. For more information, see `agent-backlog-timeout` (Bug#44262)
- When the license for MySQL Enterprise Monitor was downgraded, for example from Platimun to Gold, items that should have no longer been available were present in the Dashboard. (Bug#43823)
- The IP address of the URL used to communicate with MySQL Enterprise Service Manager would be resolved only during initial startup by an individual MySQL Enterprise Agent. This causes problems where the MySQL Enterprise Service Manager is on a virtual IP address and the DNS is used to provide failure to a new IP address. IP address caching has now been disabled, which should cause the IP address to be re-looked up and updated during a failover of the MySQL Enterprise Service Manager. (Bug#43537)
- Starting the MySQL Enterprise Agent would prevent the current session for logging out while the agent was still running. (Bug#43051)
- Queries containing `DATE_SUB` functions could show incorrect queries when normalized within Query Analyzer. (Bug#42688)
- After an upgrade from 2.0.3 to 2.0.4.7141 there was a missing label for the `agent_cpu_usage` graph.

  MySQL Enterprise Monitor was changed so that any graphs or advisors that are of a higher subscription level than currently licensed for are unscheduled and removed from the product. (Bug#42687)
- The Agent would not reconnect to a monitored database if it was started when the monitored server was down. The agent log contained the following error:

  ```
  Can't connect to MySQL server on '127.0.0.1' (0) (mysql-errno = 2003)
  ```

The agent only sent OS data to the Dashboard. Further, when the monitored server was later started, no attempts to reconnect were logged.

The problem could be worked around by restarting the agent when the monitored server was running again. (Bug#42581)

- The installer for MySQL Enterprise Agent has been updated to include support for creating the username and password required by the MySQL Enterprise Agent during operation. This process creates the user with the necessary privileges within your monitored MySQL server. For more information, see Section 5.3.1, "Creating a MySQL User Account for the Monitor Agent". (Bug#41630)

- When viewing detailed query information within the query analyzer, the incorrect 'worst' query could be shown in the detail view. (Bug#41076)

- When hovering over the heat chart, you now get a count of the number of servers affected by the corresponding rule within the tooltips, even when the list of servers, or specific server groups, are collapsed. (Bug#40823)

- In the event of a failure by MySQL Enterprise Agent to communicate with the MySQL Enterprise Service Manager, for example when the server has been restarted, it was possible for there to be gaps in the reported information. This may have shown up as gaps in the graph output. (Bug#40729)

- Queries containing an `INNER JOIN` clause could show incorrect queries when normalized within Query Analyzer. (Bug#40676)

- It was not possible to order the rows in the table within the **Manage Servers** tab. You can now sort by clicking the required column. (Bug#39457)

- Queries containing an `LIMIT` clause with value placeholders could show incorrect queries when normalized within Query Analyzer. (Bug#38892)

- The formatted query view in the pop-up within the **Query Analyzer** tab could show an empty query if the query was short and could not be formatted properly. (Bug#36885)

- When events have been raised by MySQL Enterprise Monitor, new events of the same type would not be notified until the original event had been acknowledged. This could lead to critical events being missed. In addition, when viewing existing events that have been triggered, it is impossible to know if the notified event is still causing a problem.

  The fix has been rectified through the implementation of the auto-close feature on events. With this feature enabled, events will automatically be closed once the event is not longer identified as causing a problem. (Bug#34240, Bug#23640, Bug#27132)

- If a Heat Chart rule had de-escalated and the corresponding higher event had not been closed, the Heat Chart dot link did not show the event on the Events tab. This was because the de-escalated state was shown in the Heat Chart, but an Event was not generated for it until the more serious Event was closed. (Bug#28529)

- When performing an installation of MySQL Enterprise Agent, installing multiple agents could lead to service names (Windows) and startup scripts (Unix) being overwritten, corrupted, or creating multiple entries. The installer will now ask for a different service name, or you can explicitly specify one to the installer using the `servicename` option. See `installer_servicename`. (Bug#27332)

- It was possible to import a Platinum level bundle into a MySQL Enterprise Monitor server that only had a Gold product key. (Bug#26237)

- Different Advisor jar and subscription levels could be installed at the same time, which could lead to incompatibilities and scheduling issues. In addition, you could install different Advisor jar files and corrupt the localization descriptions for the advisors. (Bug#26167, Bug#35278)

- It was not possible to assign a user a Role that had read-only permissions. (Bug#23508)