

MySQL Enterprise Monitor 2.0

MySQL Enterprise Monitor 2.0 Manual

Abstract

This manual documents the MySQL Enterprise Monitor version 2.0.

Document generated on: 2009-06-02 (revision: 15165)

Copyright 2005-2008 MySQL AB, 2009 Sun Microsystems, Inc.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms: You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Sun disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Sun Microsystems, Inc. Sun Microsystems, Inc. and MySQL AB reserve any and all rights to this documentation not expressly granted above.

Copyright © 2009 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved. U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo, MySQL Enterprise Monitor 2.0, MySQL logo(TM) and MySQL(TM) are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

Copyright © 2009 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés. L'utilisation est soumise aux termes de la Licence. Cette distribution peut comprendre des composants développés par des tierces parties. Sun, Sun Microsystems, le logo Sun, MySQL Enterprise Monitor 2.0, MySQL logo(TM) et MySQL(TM) sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Ce produit est soumis à la législation américaine en matière de contrôle des exportations et peut être soumis à la réglementation en vigueur dans d'autres pays dans le domaine des exportations et importations. Les utilisations, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exhaustive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

Table of Contents

1. Introduction	1
1.1. An Overview of the Service	1
1.1.1. The Service Architecture	2
1.1.2. Service Features	3
1.1.3. Security	4
1.2. Conventions Used in This Document	4
2. Installation and Upgrades	6
2.1. User Roles	6
2.1.1. Existing Users	6
2.1.2. User Created During Installation	7
2.1.3. Users Created on First Log-in	7
2.2. Service Manager Installation	7
2.2.1. Service Manager Installation Common Parameters	8
2.2.2. Service Manager Installation on Windows	8
2.2.3. Service Manager Installation on Mac OS X	11
2.2.4. Service Manager Installation on Unix	15
2.2.5. Starting/Stopping the MySQL Enterprise Monitor Service on Windows	17
2.2.6. Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X	18
2.2.7. MySQL Enterprise Service Manager Configuration Settings and Advisor Installation	19
2.3. Monitor Agent Installation	23
2.3.1. Creating a MySQL User Account for the Monitor Agent	23
2.3.2. Installing the Agent on Microsoft Windows	24
2.3.3. Installing the Agent on Mac OS X	28
2.3.4. Installing the Monitor Agent on Unix	34
2.3.5. Starting/Stopping the MySQL Enterprise Monitor Agent	37
2.3.6. Advanced Agent Configuration	40
2.3.7. Troubleshooting the Agent	46
2.4. Unattended Installation	47
2.4.1. Command-Line Options	47
2.4.2. Unattended Windows Installation	49
2.4.3. Unattended Unix and Mac OS X Installation	50
2.4.4. Starting the Services	50
2.5. Post-Installation Considerations	51
2.6. Upgrading, Re-Installing or Changing Your Installation	51
2.6.1. Upgrading MySQL Enterprise Monitor	52
2.6.2. Reinstalling MySQL Enterprise Monitor	59
2.6.3. Changing Your MySQL Enterprise Monitor Installation	59
2.7. Uninstalling the MySQL Enterprise Monitor	61
2.7.1. Removing the MySQL Enterprise Monitor: Windows	61
2.7.2. Removing the MySQL Enterprise Monitor: Unix and Mac OS X	63
3. Deploying MySQL Enterprise Service Manager	65
3.1. Backing up MySQL Enterprise Service Manager	65
3.2. Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0	65
3.3. Regular MySQL Enterprise Monitor Maintenance	68
3.4. Choosing Suitable MySQL Enterprise Service Manager Hardware Configurations	68
4. MySQL Enterprise Dashboard	69
4.1. The Server Tree	70
4.2. The Server Graphs and Critical Events	70
4.3. The Heat Chart	71
5. The Settings Page	74
5.1. Global Settings	74
5.2. User Preferences	76
5.3. Manage Servers	77
5.3.1. Renaming a Server	77
5.3.2. Grouping Servers	78
5.3.3. Removing a Server From the Dashboard	78
5.4. Managing Users	79
5.5. Manage Notification Groups	80
5.6. Logs	80
5.7. The Product Information Screen	81
5.7.1. Subscription Warning	81
6. The Advisors Page	83
6.1. Installing and Updating Advisors	84
6.2. Scheduling Rules	84

6.2.1. Heat Chart Notifications	86
6.3. Editing Built-in Rules	86
6.4. Creating Advisors and Rules	88
6.4.1. Creating Advisors	88
6.4.2. Overview of Rule Creation	88
6.4.3. Variables	89
6.4.4. Thresholds	89
6.4.5. Using Strings	90
6.4.6. Wiki Format	90
6.4.7. Creating a New Rule: An Example	91
6.4.8. Creating a Custom Data Collection Item	92
6.5. Disabling and Unscheduling Rules	93
6.6. Advisor Blackout Periods	93
6.6.1. Scripting Blackouts	94
7. The Events Page	95
7.1. Closing an Event	96
8. The Graphs Page	97
8.1. Displaying Graphs	97
8.2. Setting an Interval	97
8.3. Setting a Time Span	97
9. The Query Analyzer Page	99
9.1. Enabling Query Analyzer	101
9.1.1. Enabling Query Analyzer by Changing the MySQL Client Application	102
9.1.2. Enabling Query Analyzer by Changing MySQL Server	104
9.2. Getting Detailed Query Information	104
9.3. Filtering Query Analyzer Data	107
9.4. Using Query Analyzer Data	108
9.5. Troubleshooting Query Analyzer	108
9.6. Query Analyzer Settings	109
10. The Replication Page	112
10.1. Replication Page Details	112
A. MySQL Enterprise Monitor Frequently Asked Questions	114
B. Files Associated with The MySQL Enterprise Monitor	124
B.1. Log Files for the MySQL Enterprise Service Manager	124
B.2. Monitor Agent and Service Manager Installation Log Files	124
B.3. Agent Log and PID Files	125
B.4. The Management Information Base (MIB) File	125
B.5. The <code>config.properties</code> File	125
C. Error codes	127
D. MySQL Enterprise Monitor Change History	131
D.1. Changes in MySQL Enterprise Monitor 2.1.0 (Not yet released)	131
D.2. Changes in MySQL Enterprise Monitor 2.0.6 (Not yet released)	134
D.3. Changes in MySQL Enterprise Monitor 2.0.5 (18th March 2009)	135
D.4. Changes in MySQL Enterprise Monitor 2.0.4 (5th February 2009)	139
D.5. Changes in MySQL Enterprise Monitor 2.0.3 (23rd January 2009)	140
D.6. Changes in MySQL Enterprise Monitor 2.0.2 (14th January 2009)	141
D.7. Changes in MySQL Enterprise Monitor 2.0.1 (15th December 2008)	142
D.8. Changes in MySQL Enterprise Monitor 2.0.0 (11th December 2008)	142
E. The Data Collection Items Used to Create Rules	148

List of Figures

1.1. MySQL Enterprise Monitor Architecture	1
2.1. MySQL Enterprise Monitor: Installing Monitor on Windows: Language Selection	8
2.2. MySQL Enterprise Monitor: Installing Monitor on Windows: Installation Directory	9
2.3. MySQL Enterprise Monitor: Installing Monitor on Windows: Tomcat Server Options	9
2.4. MySQL Enterprise Monitor: Installing Monitor on Windows: Repository Configuration	10
2.5. MySQL Enterprise Monitor: Installing Monitor on OS X: Language Selection	12
2.6. MySQL Enterprise Monitor: Installing Monitor on OS X: Java Selection	12
2.7. MySQL Enterprise Monitor: Installing Monitor on OS X: Installation Directory	12
2.8. MySQL Enterprise Monitor: Installing Monitor on OS X: Tomcat Server Options	13
2.9. MySQL Enterprise Monitor: Installing Monitor on OS X: Repository Configuration	14
2.10. MySQL Enterprise Monitor: Initial dashboard log-in	19
2.11. MySQL Enterprise Monitor: Outgoing email settings	22
2.12. MySQL Enterprise Monitor: Installing Agent on Windows: Language Selection	24
2.13. MySQL Enterprise Monitor: Installing Agent on Windows: Installation Directory	24
2.14. MySQL Enterprise Monitor: Installing Agent on Windows: Monitored Database Information	25
2.15. MySQL Enterprise Monitor: Installing Agent on Windows: Query Analyzer Configuration	26
2.16. MySQL Enterprise Monitor: Installing Agent on Windows: MySQL Enterprise Service Manager Options	27
2.17. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Language Selection	28
2.18. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Installation Directory	29
2.19. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information	30
2.20. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information	31
2.21. MySQL Enterprise Monitor: Installing Agent on Mac OS X: Query Analyzer Configuration	32
2.22. MySQL Enterprise Monitor: Installing Agent on Mac OS X: MySQL Enterprise Service Manager Options	33
2.23. MySQL Enterprise Monitor: Server Update: Language Selection	53
2.24. MySQL Enterprise Monitor: Server Update: Previous Installation	54
2.25. MySQL Enterprise Monitor: Server Update: Backup of Previous Installation	54
2.26. MySQL Enterprise Monitor: Server Update: Completed installing files	55
2.27. MySQL Enterprise Monitor: Server Update: Final Setup	56
3.1. MySQL Enterprise Monitor: Historical Data Migration Availability	66
3.2. MySQL Enterprise Monitor: Confirming Historical Data Migration	66
3.3. MySQL Enterprise Monitor: Historical Data Migration Progress	67
4.1. MySQL Enterprise Dashboard: The Graphs screen	70
4.2. MySQL Enterprise Dashboard: The Heat Chart	71
4.3. MySQL Enterprise Dashboard: The Heat Chart legend	72
5.1. MySQL Enterprise Dashboard: Settings	74
5.2. MySQL Enterprise Dashboard: User Preferences	76
5.3. MySQL Enterprise Dashboard: Manage Servers	77
5.4. MySQL Enterprise Dashboard: Server Renaming	78
5.5. MySQL Enterprise Dashboard: Manage Users	79
5.6. MySQL Enterprise Dashboard: Manage Notification Groups	80
5.7. MySQL Enterprise Dashboard: Logs	80
6.1. MySQL Enterprise Dashboard: Scheduling Dialog	84
6.2. MySQL Enterprise Dashboard: Editing Rules	86
7.1. MySQL Enterprise Dashboard: Events screen	95
9.1. MySQL Enterprise Dashboard: Query Analyzer	100
9.2. MySQL Enterprise Dashboard: Standard agent/monitor topology	103
9.3. MySQL Enterprise Dashboard: Query Analyzer agent/monitor topology	103
9.4. MySQL Enterprise Dashboard: Canonical Query Tab for a Query	104
9.5. MySQL Enterprise Dashboard: Example Query Tab for a Query	106
9.6. MySQL Enterprise Dashboard: Explain Query Tab for a Query	107
9.7. MySQL Enterprise Dashboard: Query Analyzer Configuration	109
10.1. MySQL Enterprise Dashboard: Replication groups	112

List of Tables

6.1. MySQL Enterprise Monitor: Wiki formatting	91
B.1. MySQL Enterprise Monitor: Log file locations – Windows	124
B.2. MySQL Enterprise Monitor: Log file locations – Unix	124
B.3. MySQL Enterprise Monitor: Log file locations – Mac OS X	124
B.4. MySQL Enterprise Monitor: MIB file locations	125
B.5. MySQL Enterprise Monitor: The <code>config.properties</code> file	125
C.1. MySQL Enterprise Dashboard: GUI Error Codes and Messages	127
C.2. MySQL Enterprise Monitor: Server Codes and Messages	127
E.1. Data Collection Items	148

Chapter 1. Introduction

Note

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

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

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

MySQL Enterprise Monitor helps administrators:

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

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

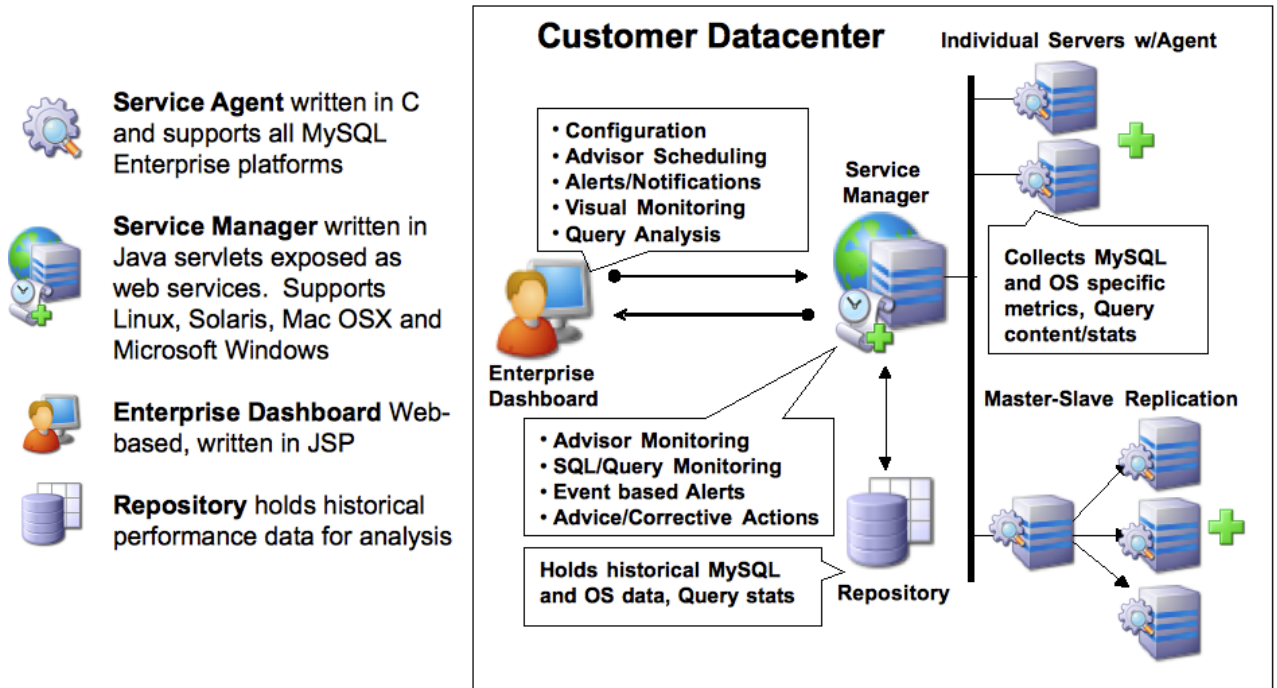
The MySQL Enterprise Monitor does this by providing a web-based interface — called the Enterprise Dashboard — that serves as the portal for viewing information about your MySQL database servers. MySQL professionals can manage all their servers by group or individually if need be.

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

1.1. An Overview of the Service

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

Figure 1.1. MySQL Enterprise Monitor Architecture



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

1.1.1. The Service Architecture

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

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

- **Monitor Agent**

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

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

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

- **Service Manager**

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

The Service Manager performs many duties including:

- Enterprise Dashboard, the main interface to the MySQL Enterprise Service Manager.
- Autodiscovery of monitored MySQL Servers.

- Creation and management of Monitor Agent tasks.
- Storage of data collections from Monitor Agents.
- Monitoring of key MySQL server and OS level metric data collections.
- Reporting MySQL best practice events and violations.
- Providing MySQL expert advice for MySQL best practice violations.
- Autodiscovery of replication topology (Not available for all subscription levels)

- **The Repository**

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

- **The Enterprise Dashboard**

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

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

1.1.2. Service Features

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

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

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

- The **MONITOR** page comprises:
 - The **SERVER TREE**: Easily navigate monitored servers
 - The **GRAPHING**: This capability is built in so you can quickly assess critical functions such as activity, performance metrics, and number of connections
 - The **HEAT CHART**: Color-coded buttons provide key operating system and database metrics
- The **ADVISORS** page

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

- The **EVENTS** page

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

- The **GRAPHS** page

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

- The **QUERY ANALYZER** page

- The **REPLICATION** page

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

- The **SETTINGS** page

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

1.1.3. Security

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

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

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

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

1.2. Conventions Used in This Document

This document uses certain typographical conventions:

- *Text in this style* is used for SQL statements; database, table, and column names; program listings and source code; and environment variables. Example: “To reload the grant tables, use the `FLUSH PRIVILEGES` statement.”
- *Text in this style* indicates input that you type in examples.
- *Text in this style* indicates the names of executable programs and scripts, examples being `mysql` (the MySQL command line client program) and `mysqld` (the MySQL server executable).
- *Text in this style* is used for variable input for which you should substitute a value of your own choosing.
- File names and directory names are written like this: “The global `my.cnf` file is located in the `/etc` directory.”
- Character sequences are written like this: “To specify a wildcard, use the `%` character.”
- *Text in this style* is used for emphasis.
- **Text in this style** is used in table headings and to convey especially strong emphasis.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Throughout this document the term ‘`Unix`’ is used to describe any Unix or Unix-like operating system. For an up-to-date list of operating systems supported by the MySQL Enterprise Monitor please see the [MySQL Enterprise web site](#).

Chapter 2. Installation and Upgrades

Note

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

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

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

Installation is a three step process:

1. Install and start the Service Manager on the monitoring system.
2. Configure the Service Manager and install the Advisors.
3. Install and start the Monitor Agent to monitor the targeted MySQL server.

Installation Requirements

The Service Manager is available for Windows, Mac OS X, and a variety of Unix and Linux operating systems.

Note

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.

To install the MySQL Enterprise Monitor on Windows requires approximately 260 MB of space and approximately 450 MB on Unix, Linux and Mac OS X. The installer checks that there is enough free space on the destination disk. However, disk space usage will increase with time since the repository stores historical data.

The minimum recommended requirements for the service manager are at least a 2GHz CPU and at least 1GB of RAM. If you are monitoring a large number of services, then there will be an increased load on the server manager. Running the service manager on a machine that is already running other tasks is only recommended if you are monitoring a small number of agents. For monitor five or more agents simultaneously, you should dedicate a machine to the process.

For more information, see [Section 3.4, “Choosing Suitable MySQL Enterprise Service Manager Hardware Configurations”](#).

The Monitor Agent is available for a wide range of operating systems. For an up-to-date list please see the [MySQL Enterprise web site](#). The agent can be used to monitor any MySQL server from version 4.0.x through 6.0.x.

2.1. User Roles

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

2.1.1. Existing Users

The **MySQL Enterprise user** – These are the credentials you use to log in to the MySQL Enterprise web site. You will need them 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 2.3.1](#),

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

2.1.2. User Created During Installation

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

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

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

2.1.3. Users Created on First Log-in

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

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

Note

The Monitor Agent has two roles in the MySQL Enterprise Monitor; it must have access to the dashboard and to the MySQL server it is monitoring. For a description of the agent as a MySQL user see [Section 2.1.1, “Existing Users”](#).

2.2. Service Manager Installation

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

- Apache Tomcat
- MySQL Server
- Java VM

Note

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

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

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

Warning

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

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

2.2.1. Service Manager Installation Common Parameters

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

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

- **Tomcat Server Options**

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

Note

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

- Tomcat Shutdown port —the port used by the management scripts that is used to shut the Tomcat server down when you need to stop the Service Manager. The default value is 18005.
- Tomcat SSL Port — the standard port used to connect to the Service Manager when you want to use Secure Sockets Layer (SSL) encrypted communication. The default value is 18443.

- **Repository Configuration (MySQL Server)**

- Repository Username — the user name created and used to store information within the MySQL server to hold the information used by the Service Manager. In normal use, you should not need to use or modify this information, but it may be required if you have a support issue. The default value is `service_manager`.
- Repository User password — the password to be used for the Repository Username. This should be set to a secure password so that the repository data is secure.

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

Caution

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

2.2.2. Service Manager Installation on Windows

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

Note

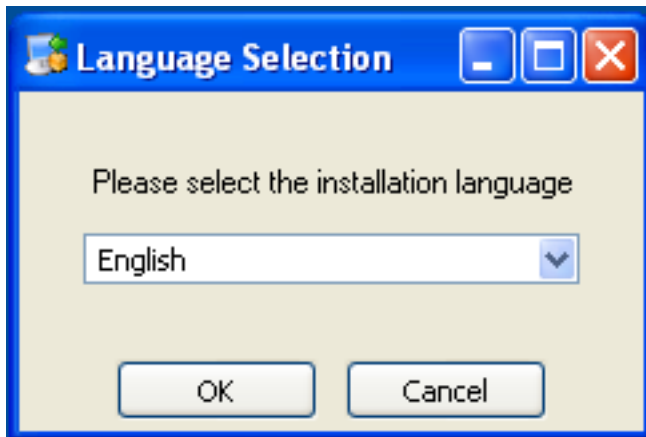
In order to install the Service Manager as a Windows service, you must do the installation as a privileged user.

On Windows Vista, 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).

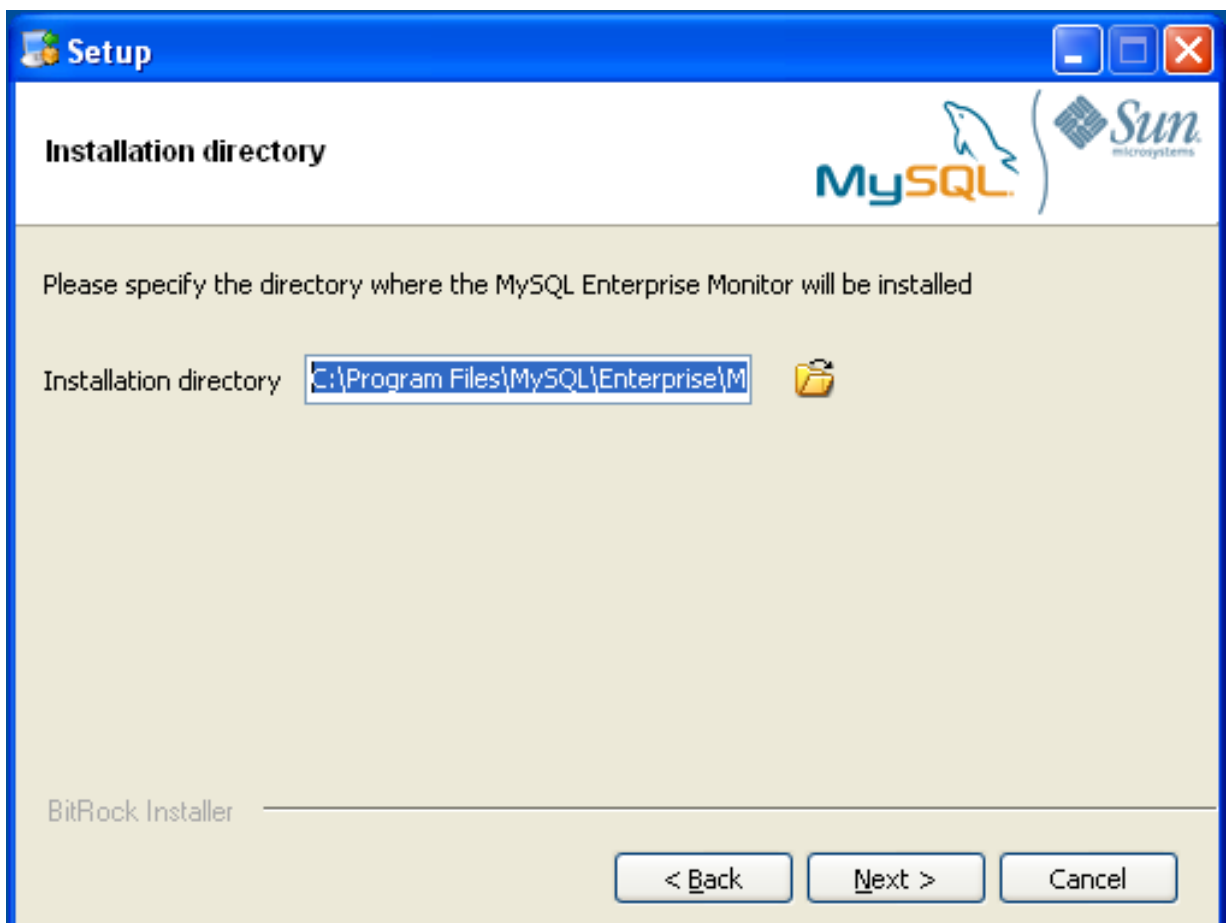
1. Double click on the MySQL Monitor installer. You should be presented with the Language Selection prompt. Select the language to use for the installer and then click OK.

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



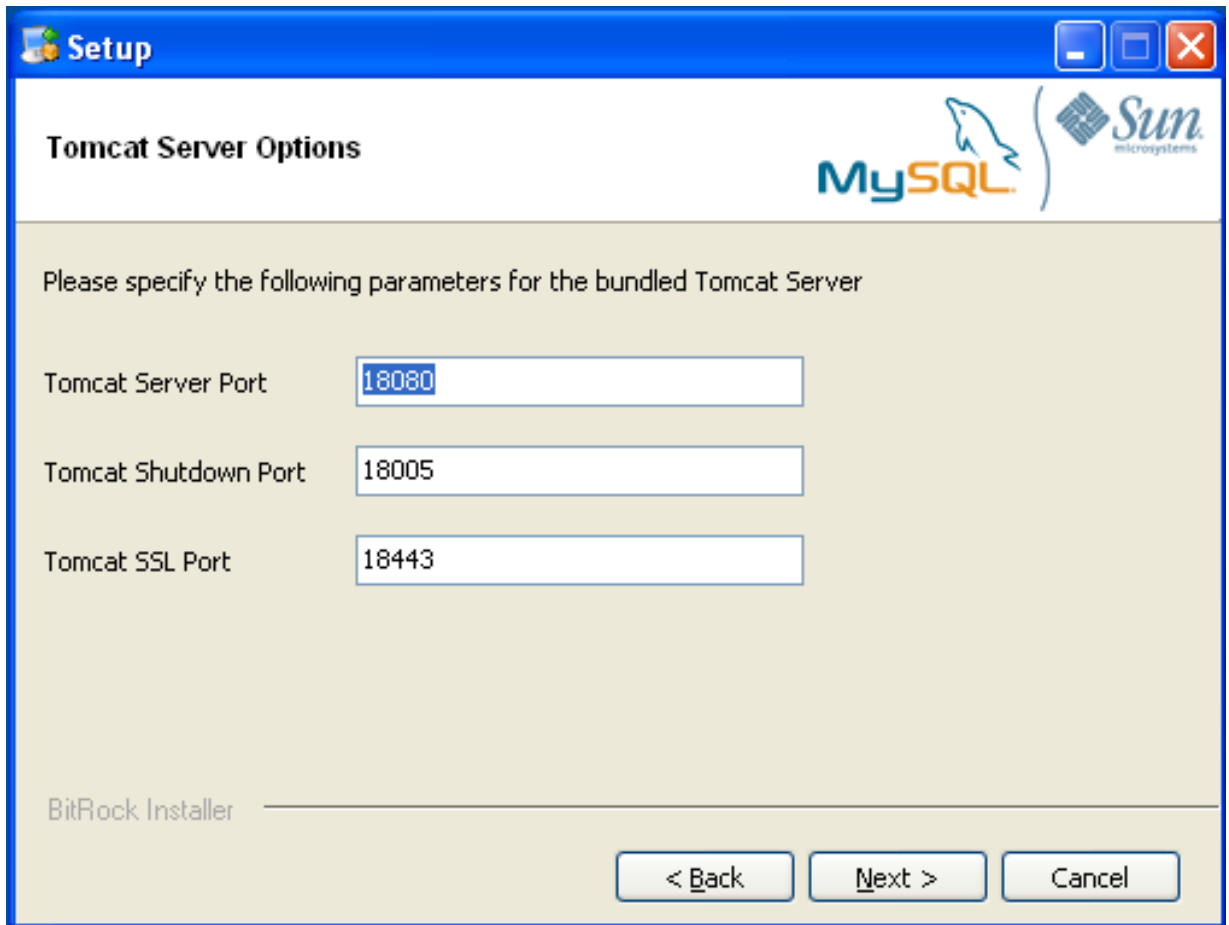
2. With the installation language selected, the remainder of the installation sets up the installation location and the main configuration parameters required by MySQL Enterprise Service Manager. Click NEXT to continue.
3. Select the installation directory where you want the MySQL Enterprise Service Manager components installed. By default on Windows the directory is `C:\Program Files\MySQL\Enterprise\Monitor`. You click the button next to the installation directory field to select a directory using the File chooser, or type the directory manually. Click NEXT to continue.

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



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

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

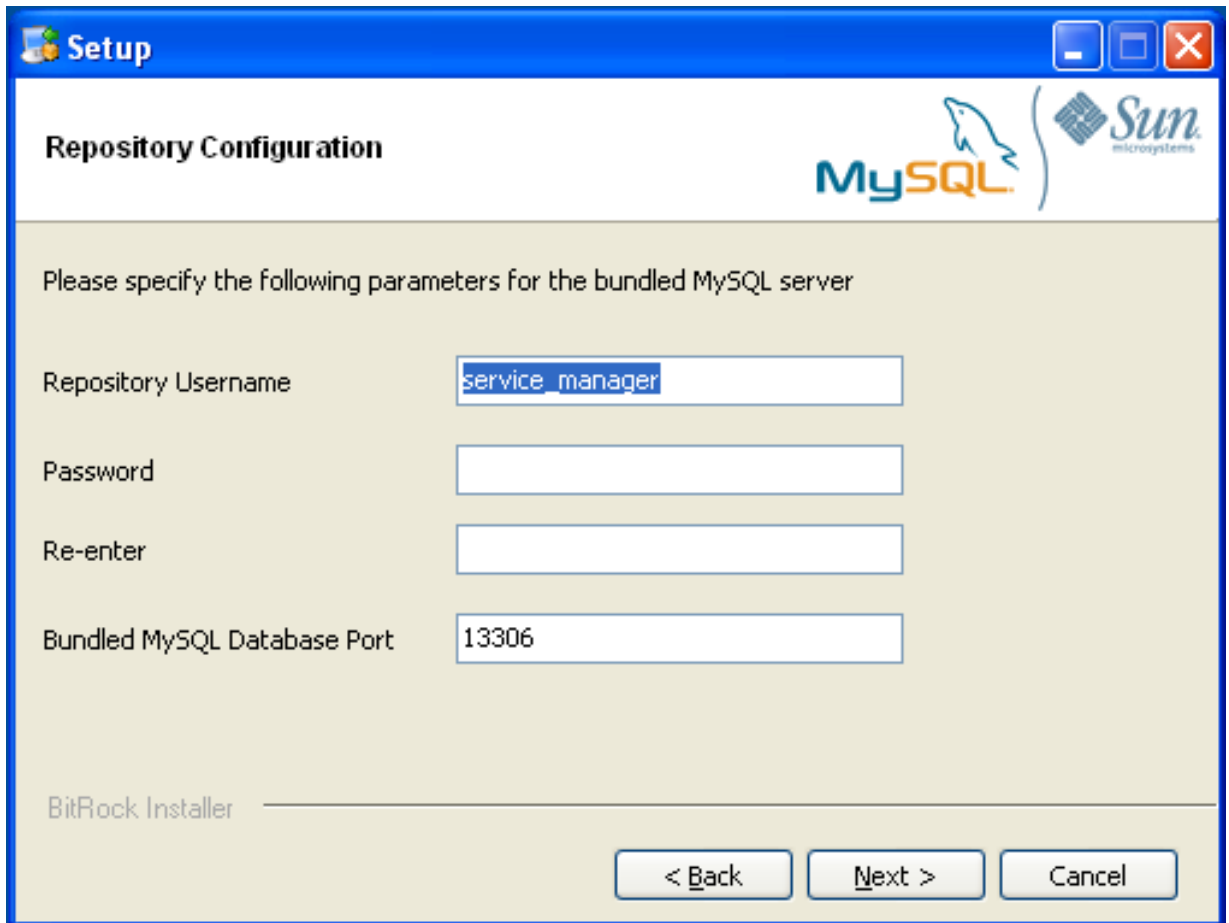


5. Configure the repository settings, setting the user name, password and port used to communicate with the bundled MySQL server that will be used to store the information and statistics for your installation. For more information, see [Section 2.2.1, "Service Manager Installation Common Parameters"](#).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

Note

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

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



6. You will be provided with information and a warning about the configuration options and how they are stored in the `configuration_report.txt` file, and its location. Take a note of the full path to this file in case you need to look up the information later. Click NEXT to continue.
7. You should now be prompted to start the installation process. Click NEXT to continue.
8. Once the installation has been completed, you will be provided with the information on how to uninstall MySQL Enterprise Service Manager. Click NEXT to continue.
9. To complete the installation and set up your MySQL Enterprise Service Manager, you will need to login to the Dashboard. You can do this automatically by checking the box on the final window before clicking FINISH. This checkbox is selected by default. If you do not want to run the Dashboard at this time, uncheck the box and click FINISH.

For instructions on starting the MySQL Enterprise Monitor services under Windows, see [Section 2.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows”](#).

2.2.3. Service Manager Installation on Mac OS X

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

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

For reasons of 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.

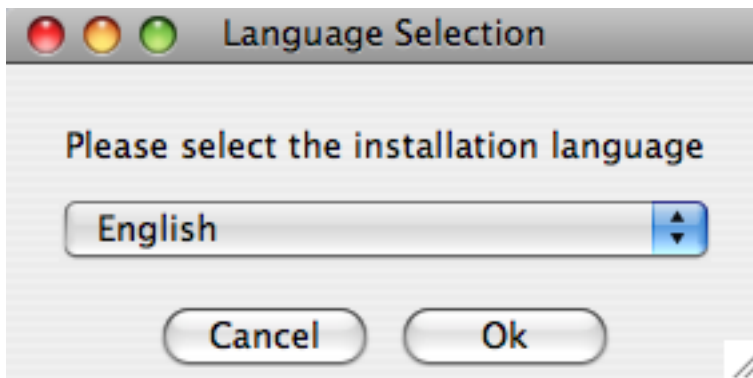
If you have changed the default you can reset it or you may install the MySQL Enterprise Service Manager in `text` mode, setting

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 2.2.4, “Service Manager Installation on Unix” with the minor differences noted above.

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

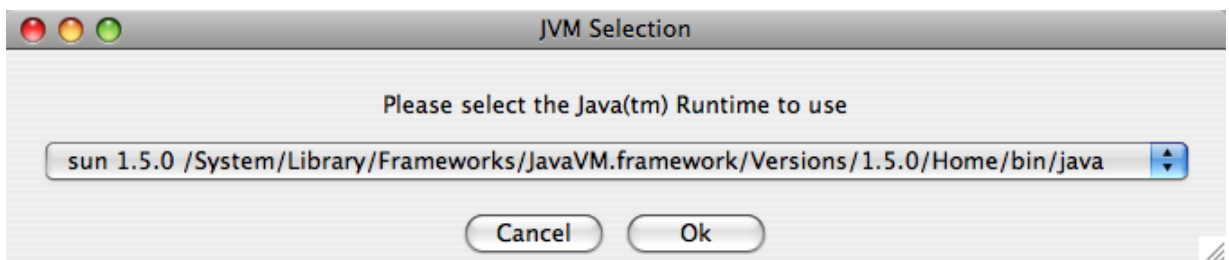
1. Double click on the MySQL Monitor installer. You should be presented with the Language Selection prompt. Select the language to use for the installer and then click OK.

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



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

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



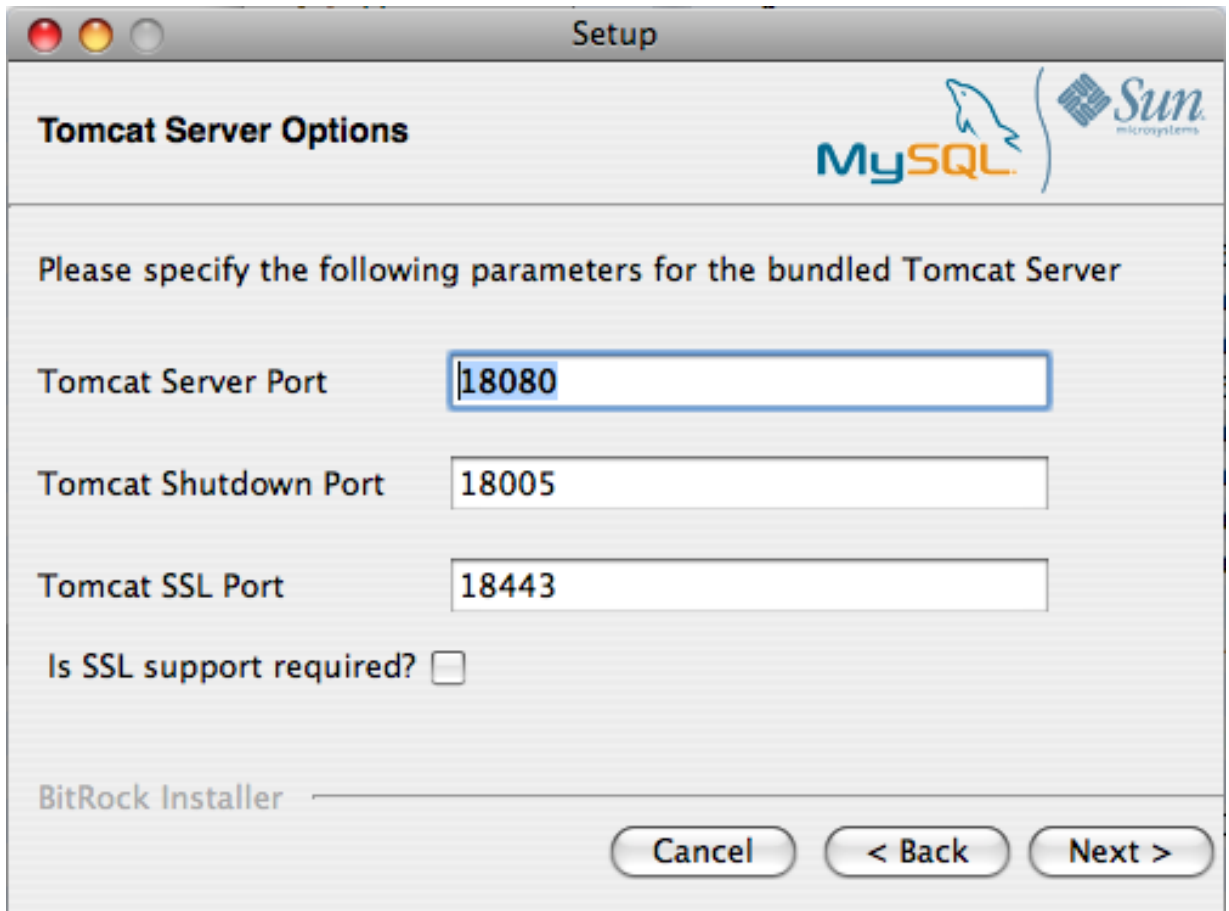
3. With the installation language and Java version selected, the remainder of the installation sets up the installation location and the main configuration parameters required by MySQL Enterprise Service Manager. Click NEXT to continue.
4. Select the installation directory where you want the MySQL Enterprise 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 2.7. MySQL Enterprise Monitor: Installing Monitor on OS X: Installation Directory



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

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



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

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

Setup

Repository Configuration

Please specify the following parameters for the bundled MySQL server

Repository Username:

Password:

Re-enter:

Bundled MySQL Database Port:

BitRock Installer

Cancel < Back Next >

7. You will be provided with information and a warning about the configuration options and how they are stored in the `configuration_report.txt` file, and its location. Take a note of the full path to this file in case you need to look up the information later. Click NEXT to continue.
8. You should now be prompted to start the installation process. Click NEXT to continue.
9. Once the installation has been completed, you will be provided with the information on how to uninstall MySQL Enterprise Service Manager. Click NEXT to continue.
10. To complete the installation and set up your MySQL Enterprise Service Manager, you will need to login to the Dashboard. You can do this automatically by checking the box on the final window before clicking FINISH. This checkbox is selected by default. If you do not want to run the Dashboard at this time, uncheck the box and click FINISH.

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

2.2.4. Service Manager Installation on Unix

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

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

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

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

1. Begin installation by typing:

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

2. First choose the language for the installation:

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

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

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

```
Please specify the directory where the MySQL Enterprise Service Manager
will be installed.
Installation directory [/opt/mysql/enterprise/monitor/]:
```

4. Now set the Tomcat Server options. For more details on these parameters, see [Section 2.2.1, “Service Manager Installation Common Parameters”](#).

```
-----
Tomcat Server Options
Please specify the following parameters for the bundled Tomcat Server
Tomcat Server Port [18080]:
Tomcat Shutdown Port [18005]:
Tomcat SSL Port [18443]:
```

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

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

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

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

```
-----
Repository Configuration
Please specify the following parameters for the bundled MySQL server
Repository Username [service_manager]:
Password :
Re-enter :
Bundled MySQL Database Port [13306]:
```

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

```
-----
Configuration Report
Note:
The settings you specified will be saved here:
/opt/mysql/enterprise/monitor/configuration_report.txt
```

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

```
Press [Enter] to continue :
```

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

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

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

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

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

```
Completed installing files
```

```
Setup has completed installing MySQL Enterprise files on your computer
```

```
Uninstalling the MySQL Enterprise files can be done by invoking:
```

```
/opt/mysql/enterprise/monitor/uninstall
```

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

```
Press [Enter] to continue :
```

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

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

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

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

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

```
Press [Enter] to continue :
```

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

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

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

- MySQL Enterprise Tomcat
- MySQL Enterprise MySQL

You can stop or start the services from the Microsoft Management Console Services window. Look for the [MySQL Enterprise Tomcat](#) and the [MySQL Enterprise MySQL](#) entries.

Note

On Windows Vista, 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 restric-

tion applies to using the menu options discussed in the following and to starting the services from the command line. To open an administrator `cmd` window right click the `cmd` icon and choose the `RUN AS ADMINISTRATOR` menu option.

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

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

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

```
shell> sc start MySQLEnterpriseTomcat
```

or:

```
shell> net start MySQLEnterpriseTomcat
```

To stop this service type:

```
shell> sc stop MySQLEnterpriseTomcat
```

or:

```
shell> net stop MySQLEnterpriseTomcat
```

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

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

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

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

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

Configuration of the dashboard begins immediately after the Service Manager is installed. To continue a Windows installation skip the next section and go to [Section 2.2.7, "MySQL Enterprise Service Manager Configuration Settings and Advisor Installation"](#).

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

The services incorporated into the MySQL Enterprise Service Manager are:

- The MySQL Server
- The Apache/Tomcat Server

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

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

Executing this script produces the following output:

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

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

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

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

To start the service:

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

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

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

To stop the service:

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

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

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

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

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

2.2.7. MySQL Enterprise Service Manager Configuration Settings and Advisor Installation

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

If you installed the Service Manager using a graphical interface, you have the option of launching the dashboard on the final installation screen (as long as the LAUNCH MYSQL ENTERPRISE MONITOR NOW checkbox is checked).

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

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

2.2.7.1. Initial Dashboard Log-in

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

Figure 2.10. MySQL Enterprise Monitor: Initial dashboard log-in

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

Enterprise Credentials and Subscription Information

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

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

Email Address (MySQL Enterprise Login)

Enterprise Password (MySQL Enterprise Password)

Confirm Password

- OR -

MySQL Enterprise Product Key

Advisor .jar File (recommended)

Create Administrator	Configure Agent Credentials
Username <input type="text" value="admin"/>	Username <input type="text" value="agent"/>
Password <input type="password"/>	Password <input type="password"/>
Confirm Password <input type="password"/>	Confirm Password <input type="password"/>

Use this screen to perform the following tasks:

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

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

Note

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

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

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

In the **CONFIGURE AGENT CREDENTIALS** section of this screen enter the credentials for the agent. This is the [agent user](#) also described in [Section 2.1.3, “Users Created on First Log-in”](#). The agent needs to log in in order to report its findings. Make note of the agent's credentials; this information is required when installing the agent.

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

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

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

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

2.2.7.2. Setting the Timezone and Locale

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

Warning

It is especially important that the time zone be set correctly as this may also affect the way the graphs display. For this reason, also ensure that the time reported by the operating system is correct. To change the time zone or locale see [Section 5.2, “User Preferences”](#).

The locale chosen determines the user's default language when logging in to the Dashboard. Note that this will override the default browser settings whenever this specific user logs in.

After specifying your time zone and locale, the dashboard opens on the [Monitor](#) page. For a detailed examination of the [Monitor](#) Screen see, [Chapter 4, MySQL Enterprise Dashboard](#).

2.2.7.3. Installing Advisors After Initial Log-in

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

If you did not install the Advisors when you first logged in to the MySQL Enterprise Dashboard, open the dashboard, find the [Advisors](#) tab, and choose the [Check for Updates](#) link. Doing this downloads the latest version of the Advisors from the MySQL Enterprise web site. In order to install the advisors in this fashion you must specify your MySQL Enterprise credentials. Find instructions for doing this in [Section 5.1, “Global Settings”](#).

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

2.2.7.4. Upgrading and Updating Advisors

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

Note

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

2.2.7.5. Outgoing Email Settings

Alert notification 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 can configure the email settings. These settings apply to the currently logged-in user.

Find the [Outgoing Email Settings](#) on the left of this page.

Figure 2.11. MySQL Enterprise Monitor: Outgoing email settings

Outgoing Email Settings

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

SMTP Server

SMTP Server Login

Disable JavaMail TLS/SSL

Update Password On Save

SMTP Server Password **Confirm Password**

On Save, Send Test Email Message to (optional)

Ensure that the [Enable Email Notifications](#) checkbox is checked and enter information as appropriate.

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

Note

An email server must be available for sending email alerts.

The SMTP client uses Transport Layer Security (TLS) if the SMTP server supports it.

If your SMTP server incorrectly indicates that it supports TLS, check the **DISABLE JAVAMAIL TLS/SSL** check box.

The email settings page is dealt with in more detail in [Chapter 5, The Settings Page](#).

2.3. Monitor Agent Installation

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

Warning

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

2.3.1. Creating a MySQL User Account for the Monitor Agent

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

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

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

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

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

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

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

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

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

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

Note

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

In a typical configuration, the agent runs on the same machine as the MySQL server it is monitoring so the host name will be `loc-`

localhost. However, this will change if the agent is running on a machine other than the one that hosts the monitored MySQL server. In this case, change *localhost* to the appropriate value. For more information about remote monitoring see [Section 2.3.6.4, “Configuring an Agent to Monitor a Remote MySQL Server”](#).

2.3.2. Installing the Agent on Microsoft Windows

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

Note

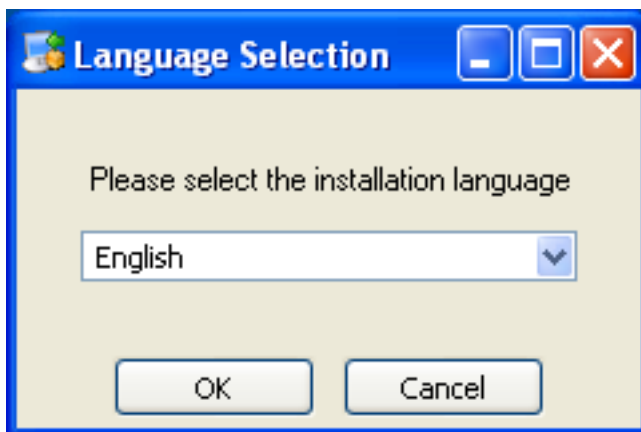
In order to install the agent as a Windows service, you must do the installation as a privileged user.

On Windows Vista, if user account control is on, an operating system dialog box requests confirmation of the installation.

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

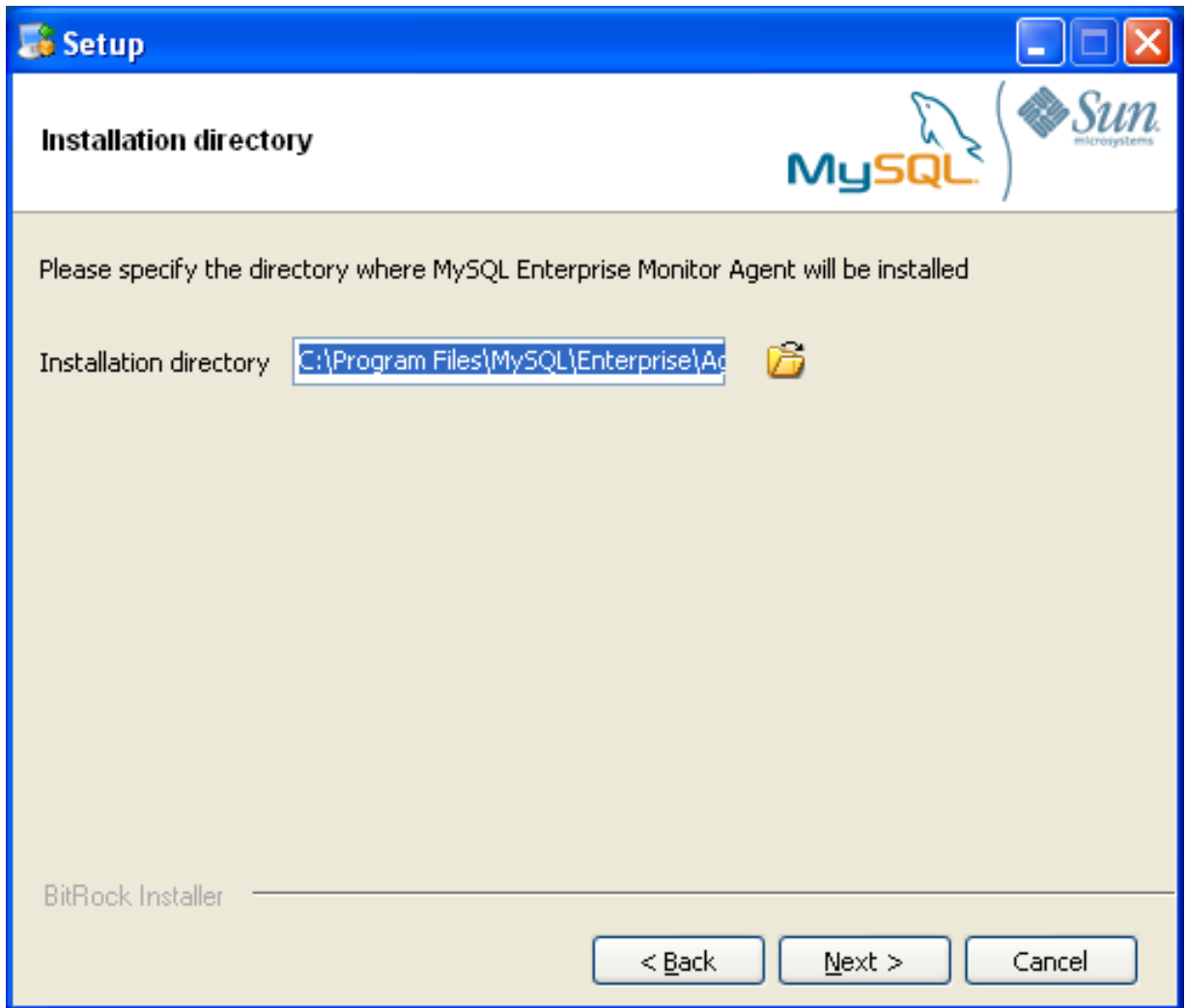
1. First, select the language for the MySQL Enterprise Monitor Agent installation. Click OK to continue installation.

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



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

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



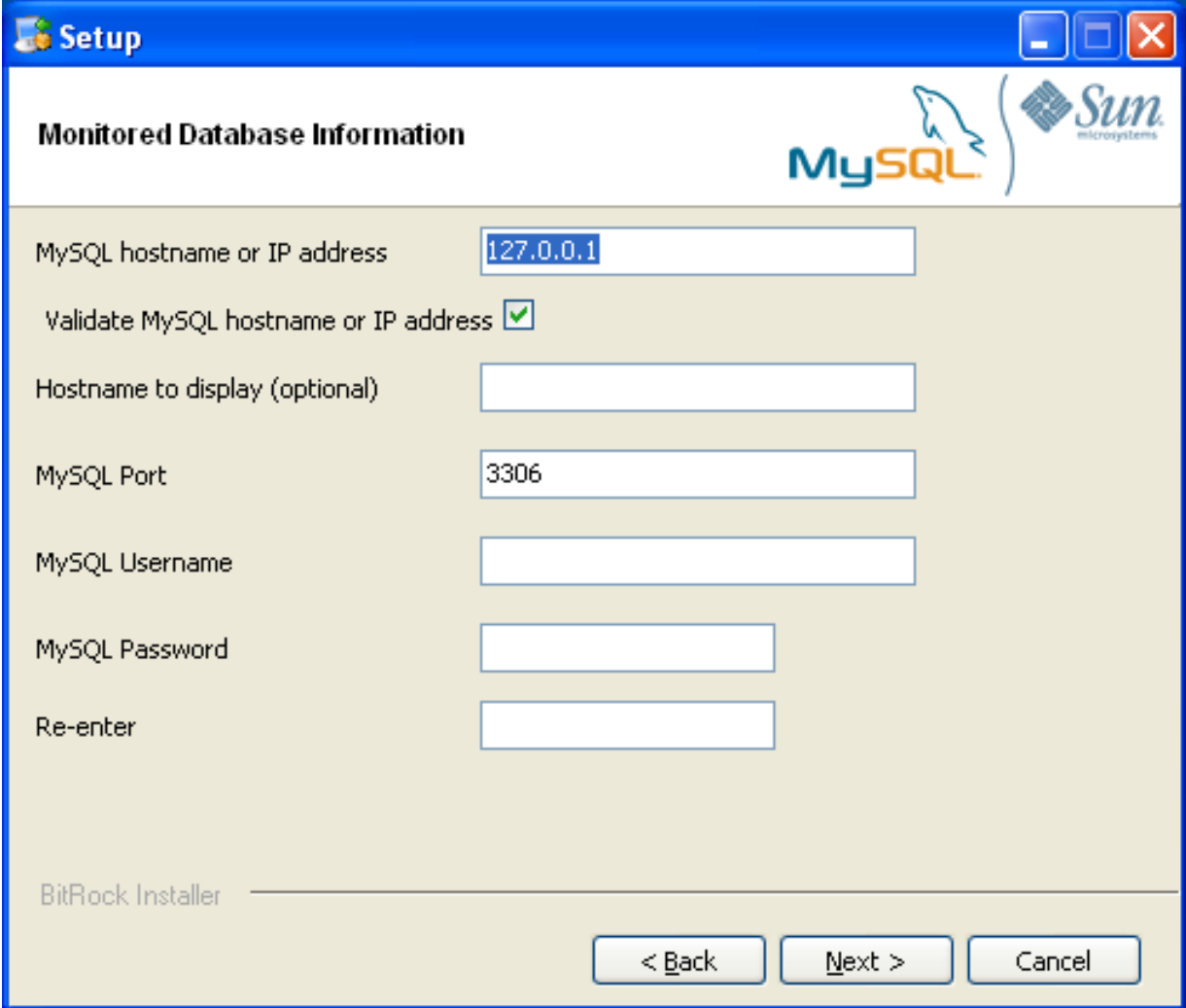
4. You need to specify the information about the MySQL server that you want to monitor. You must enter the IP address or host name of the host you want to monitor, and the port, user name and password that you will use to connect to the MySQL server. If you want to confirm that the MySQL server is currently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

Note

Currently, on Windows, the service 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` then the server will only listen for connections on the IP address specified, that is, the IP address of the MySQL server. If the service agent has been started using TCP/IP networking and the default address of 127.0.0.1 it will not be able to connect to the server to be monitored. Also, if "localhost" is specified as the host name during agent configuration, a connection will not be established, as the server will be listening for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

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



The screenshot shows a Windows-style window titled "Setup" with a blue header bar. The main content area is titled "Monitored Database Information" and features the MySQL and Sun Microsystems logos in the top right. The configuration fields are as follows:

MySQL hostname or IP address	<input type="text" value="127.0.0.1"/>
Validate MySQL hostname or IP address	<input checked="" type="checkbox"/>
Hostname to display (optional)	<input type="text"/>
MySQL Port	<input type="text" value="3306"/>
MySQL Username	<input type="text"/>
MySQL Password	<input type="password"/>
Re-enter	<input type="password"/>

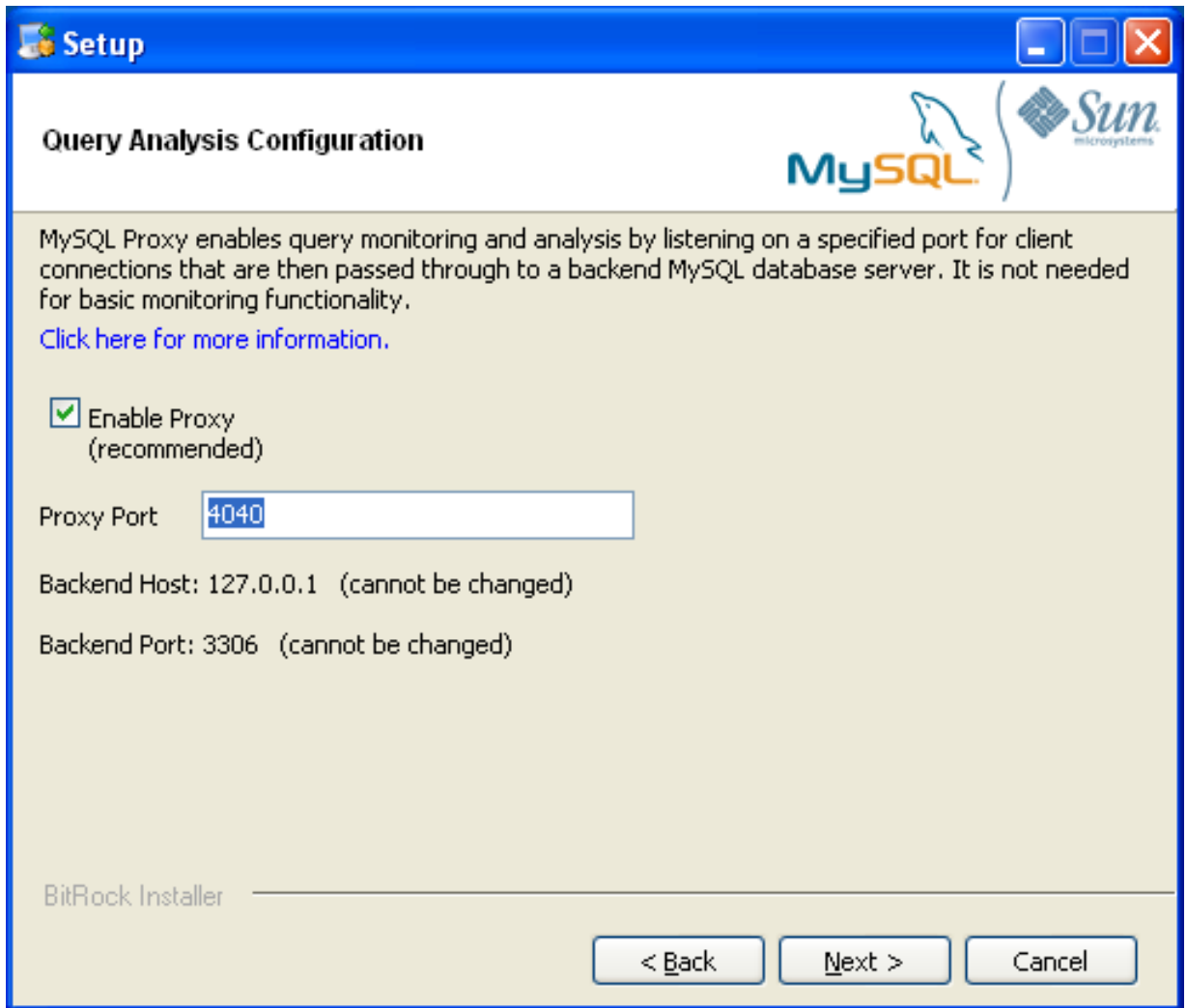
At the bottom left, it says "BitRock Installer". At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

Click NEXT to continue the installation.

5. If you want to use Query Analyzer, then you need to enable the MySQL Enterprise Monitor Agent Proxy. The Proxy is enabled by default. If you disable the Proxy during installation, you will need to enable it later before you are able to use Query Analysis. For more information on Query Analyzer, see [Chapter 9, The Query Analyzer Page](#).

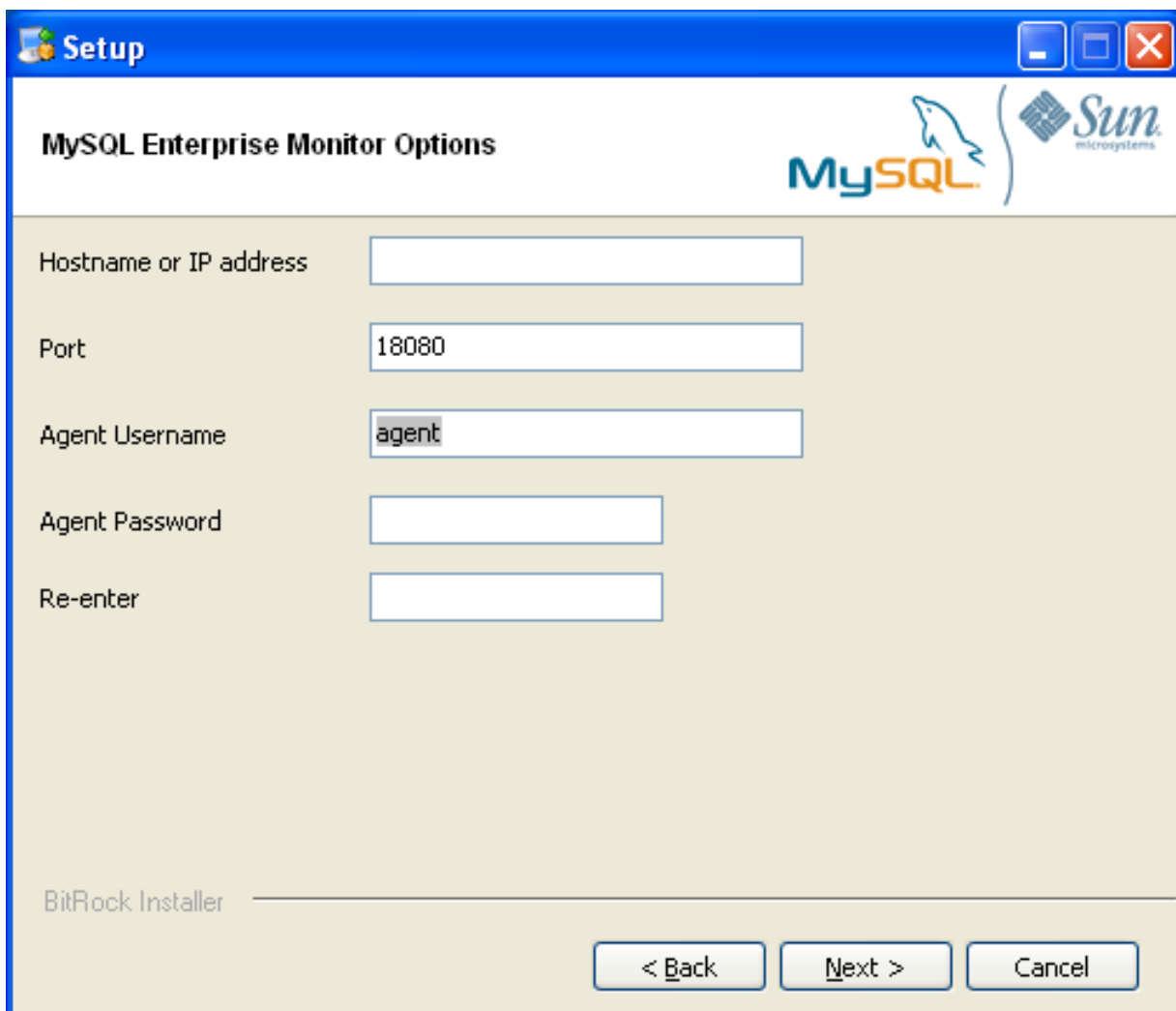
When Proxy is enabled, MySQL Enterprise Monitor Agent listens on a network port for client applications, and forwards the connections to the backend MySQL server. You can change the port number that MySQL Enterprise Monitor Agent listens for connections. The default port is 4040.

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



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

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



7. You will be provided with a Configuration Report containing the information that you have entered during the installation. Check the information provided in the report. If you see a problem, use BACK to go back to the configuration screen and change the information. If the information is correct, click NEXT to continue.
8. You are given a final opportunity to change the installation parameters. Click NEXT to start the installation process.
9. Once the agent has been installed, you will get a confirmation message. Click NEXT to finalize the installation.
10. You can start the MySQL Enterprise Monitor Agent automatically now the installation has been completed. To allow the agent to be started, leave the checkbox selected. To start the agent separately, uncheck the checkbox. Click FINISH to exit the installation.

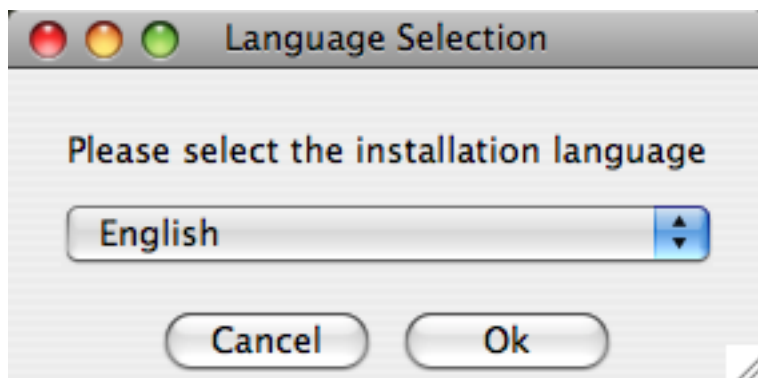
Once the Monitor Agent is installed, it needs to be started. For information on how to start and stop the Agent, see [Section 2.3.5.1, "Starting/Stopping the Agent on Windows"](#).

2.3.3. Installing the Agent on Mac OS X

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

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

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



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

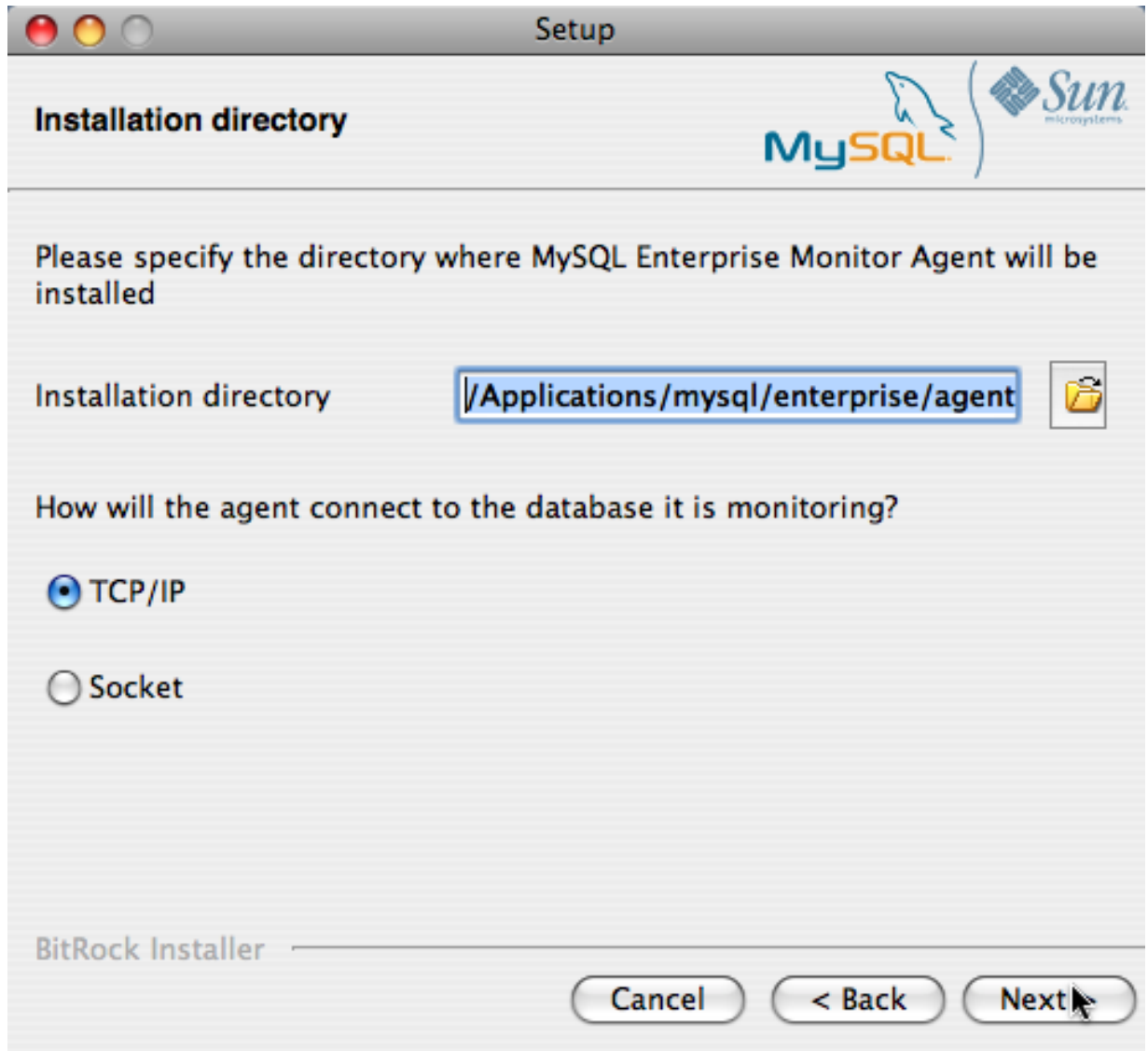
Note

The service 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 host-name “localhost” is specified.

If the MySQL server you wish to monitor has been started with the `--skip-networking` command option then you will not be able to connect to it via TCP/IP, as the server will not listen for TCP/IP connections. In this case the service agent will need to be configured to use the MySQL socket. This can be done during installation by selecting “socket” rather than “TCP/IP” and then specifying the MySQL socket name. This can also be configured after installation by editing the `agent-instance.ini` configuration file, for further information on this refer to [Section 2.3.6.2, “MySQL Server \(agent-instance.ini\) Configuration”](#).

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

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



4. You need to specify the information about the MySQL server that you want to monitor. The configuration information you enter will depend on the connection method selected in the previous screen.
 - If you chose TCP/IP as the connection method, you must enter the IP address or host name of the host you want to monitor, and the port, user name and password that you will use to connect to the MySQL server. If you want to confirm that the MySQL server is currently reachable using the information, ensure that the **VALIDATE MYSQL HOST NAME OR IP ADDRESS** checkbox is selected.

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

Setup

Monitored Database Information

MySQL

Sun
microsystems

IMPORTANT: The agent user account specified below requires special MySQL privileges.
[Click here for more information.](#)

MySQL hostname or IP address

Validate MySQL hostname or IP address

MySQL Port

MySQL Username

MySQL Password

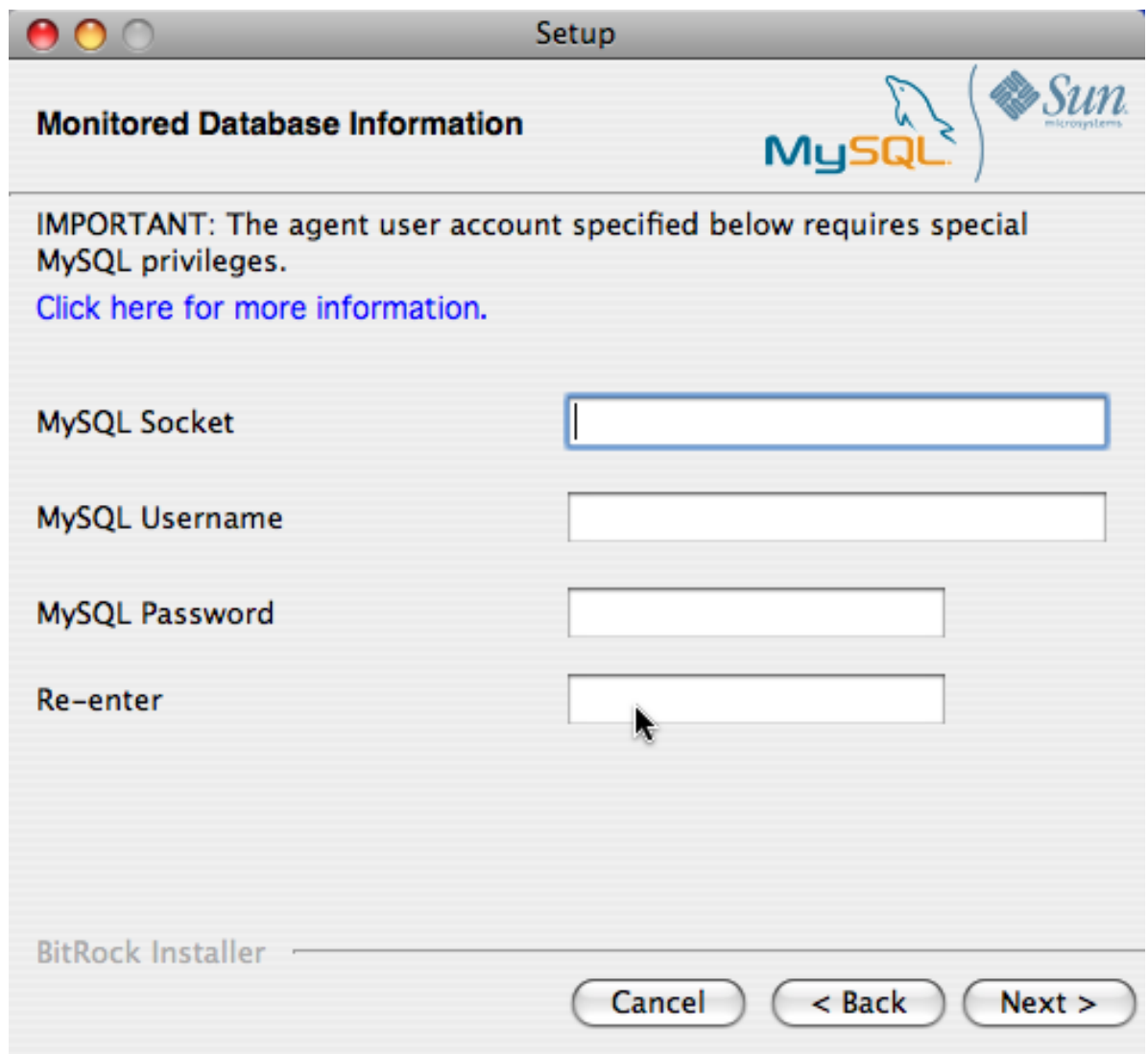
Re-enter

BitRock Installer

Cancel < Back Next >

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

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

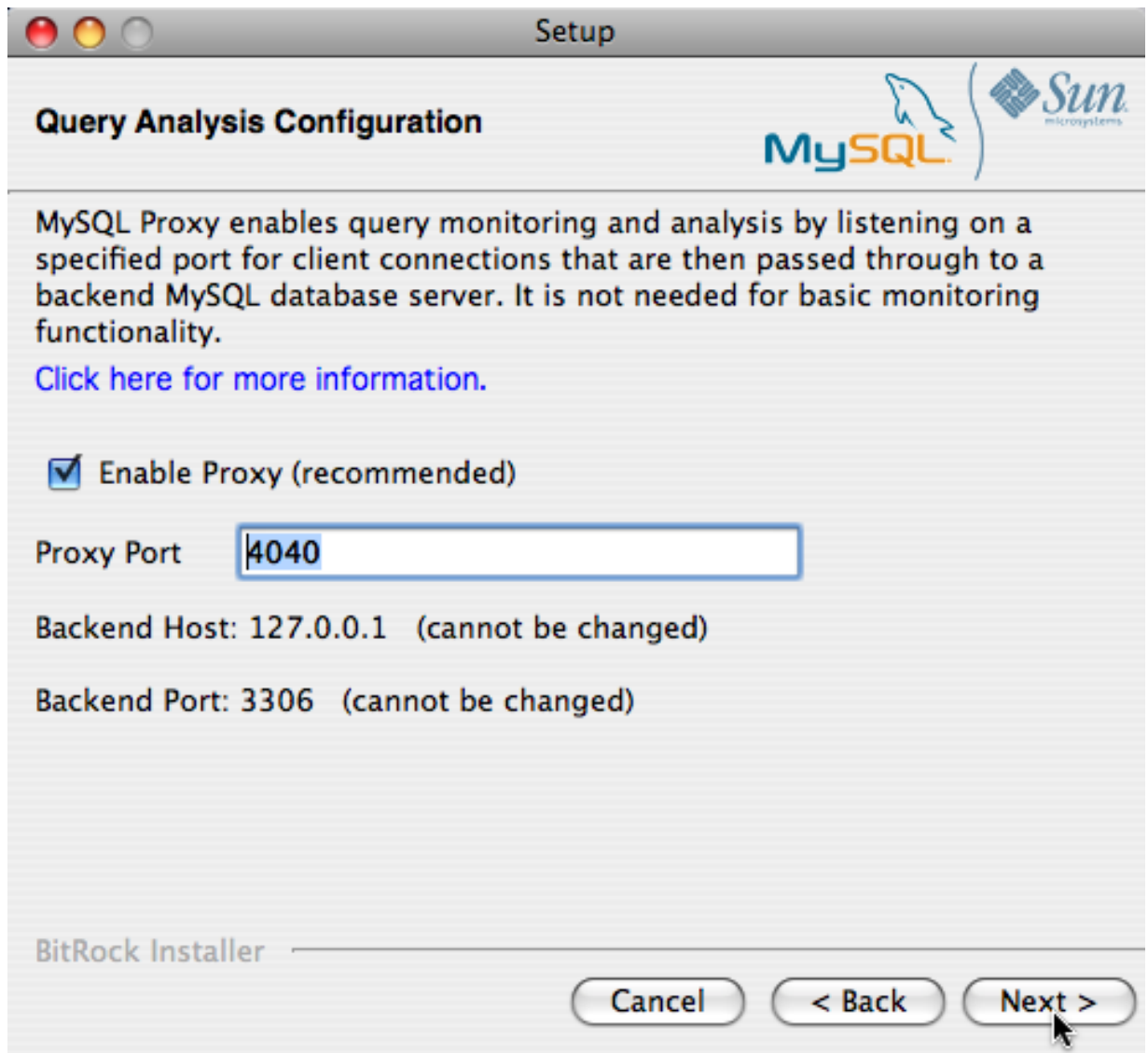


Click NEXT to continue the installation.

5. If you want to use Query Analyzer, then you need to enable the MySQL Enterprise Monitor Agent Proxy. The Proxy is enabled by default. If you disable the Proxy during installation, you will need to enable it later before you are able to use Query Analysis. For more information on Query Analyzer, see [Chapter 9, The Query Analyzer Page](#).

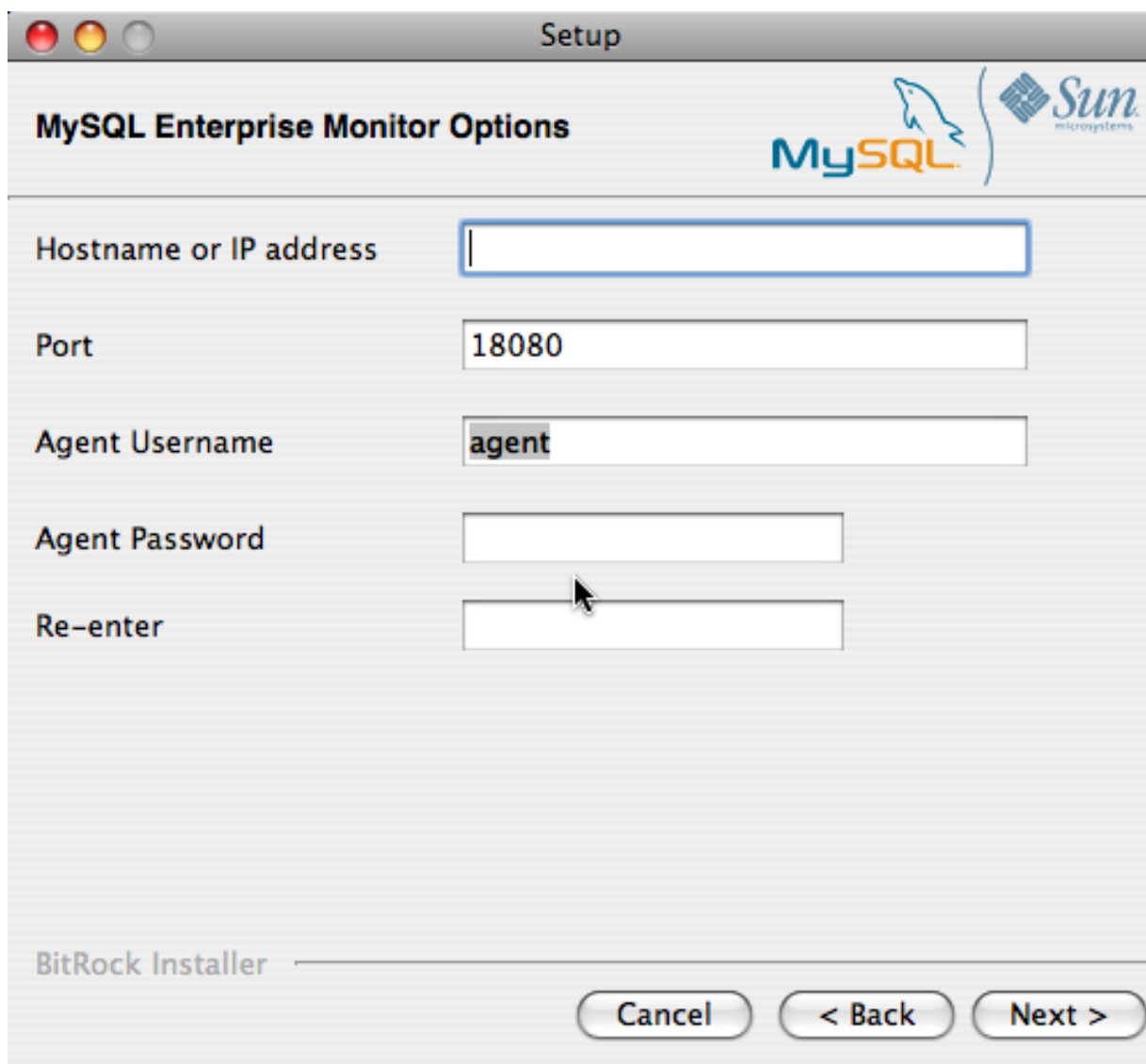
When Proxy is enabled, MySQL Enterprise Monitor Agent listens on a network port for client applications, and forwards the connections to the backend MySQL server. You can change the port number that MySQL Enterprise Monitor Agent listens for connections. The default port is 4040.

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



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

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



Setup

MySQL Enterprise Monitor Options

MySQL Sun
microsystems

Hostname or IP address

Port

Agent Username

Agent Password

Re-enter

BitRock Installer

Cancel < Back Next >

7. You will be provided with a Configuration Report containing the information that you have entered during the installation. Check the information provided in the report. If you see a problem, use BACK to go back to the configuration screen and change the information. If the information is correct, click NEXT to continue.
8. You are given a final opportunity to change the installation parameters. Click NEXT to start the installation process.
9. Once the agent has been installed, you will get a confirmation message. Click NEXT to finalize the installation.
10. You can start the MySQL Enterprise Monitor Agent automatically now the installation has been completed. To allow the agent to be started, leave the checkbox selected. To start the agent separately, uncheck the checkbox. Click FINISH to exit the installation.

Once the Monitor Agent is installed, it needs to be started. For information on how to start and stop the Agent, see [Section 2.3.5.2, "Starting/Stopping the Agent on Mac OS X"](#).

2.3.4. Installing the Monitor Agent on Unix

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

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

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

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

Note

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

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

Begin installation from the command line by typing:

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

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

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

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

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

```
-----
Welcome to the MySQL Enterprise Monitor Agent Setup Wizard.
-----
Please specify the directory where MySQL Enterprise Monitor Agent will be installed
Installation directory [/opt/mysql/enterprise/agent]:
```

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

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

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

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

```
-----
Monitored Database Information
IMPORTANT: The agent user account specified below requires special MySQL privileges.
Visit the following URL for more information:
https://enterprise.mysql.com/docs/monitor/2.0/en/mem-install.html#mem-agent-rights
MySQL hostname or IP address [127.0.0.1]:

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

MySQL Port [3306]:
```

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

```
-----
Monitored Database Information
IMPORTANT: The agent user account specified below requires special MySQL privileges.
```

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

Note

The service 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 host-name “localhost” is specified.

If the MySQL server you wish to monitor has been started with the `--skip-networking` command option then you will not be able to connect to it via TCP/IP, as the server will not listen for TCP/IP connections. In this case the service agent will need to be configured to use the MySQL socket. This can be done during installation by selecting “socket” rather than “TCP/IP” and then specifying the MySQL socket name. This can also be configured after installation by editing the `agent-instance.ini` configuration file, for further information on this refer to [Section 2.3.6.2, “MySQL Server \(agent-instance.ini\) Configuration”](#).

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

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

```
MySQL Username []: service_agent
MySQL Password :
Re-enter :
```

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

```
-----
Query Analyzer Configuration
MySQL Proxy enables query monitoring and analysis by listening on a specified port for client connections that are
Click here for more information.
[Y/n]:
Enable Proxy (recommended) [Y/n]:

Proxy Port [4040]:
Backend Host: 127.0.0.1 (cannot be changed)
Backend Port: 3306 (cannot be changed)
```

For more information on enabling Query Analyzer if you disabled it during installation, see [Chapter 9, The Query Analyzer Page](#).

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

```
-----
MySQL Enterprise Monitor Options

Hostname or IP address []: 192.168.0.197
Tomcat Server Port [18080]:
Tomcat SSL Port [18443]:
```

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

```
Use SSL? [y/N]:
Agent Username [agent]:
Agent Password :
Re-enter :
-----
```

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

```
Here are the settings you specified:
Installation directory: /opt/mysql/enterprise/agent
Monitored MySQL Database:
-----
Hostname or IP address: 127.0.0.1
Port: 3306
MySQL username: mysql_user
MySQL password: password
Query Analyzer Configuration
-----
Proxy Enabled: yes
Proxy Port: 4040
MySQL Enterprise Manager:
-----
Hostname or IP address: 192.168.0.197
Tomcat Server Port: 18080
Tomcat SSL Port: 18443
Use SSL: 0
Agent username: agent
Press [Enter] to continue :
-----
Setup is now ready to begin installing MySQL Enterprise Monitor Agent on your computer.
Do you want to continue? [Y/n]: y
```

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

```
-----
Please wait while Setup installs MySQL Enterprise Monitor Agent on your computer.
Installing
0% _____ 50% _____ 100%
#####
-----
Info to start MySQL Agent
The MySQL agent was successfully installed. To start the MySQL Agent please
invoke:
/opt/mysql/enterprise/agent/etc/init.d/mysql-monitor-agent start
Press [Enter] to continue :
-----
Setup has finished installing MySQL Enterprise Monitor Agent on your computer.
```

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

For information on starting the agent, see [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#).

2.3.5. Starting/Stopping the MySQL Enterprise Monitor Agent

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

■ Warning

If you are using Query Analyzer, then turning off the agent will prevent your applications from communicating with the MySQL server. See [Chapter 9, The Query Analyzer Page](#).

2.3.5.1. Starting/Stopping the Agent on Windows

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

Note

On Windows Vista, starting the agent requires administrative privileges — you must be logged in as an administrator. To start or stop the agent right click the menu item and choose the [RUN AS ADMINISTRATOR](#) menu option. The same restriction applies to starting the agent from the command line. To open an administrator [cmd](#) window right-click the [cmd](#) icon and choose the [RUN AS ADMINISTRATOR](#) menu option.

Warning

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

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

```
shell> sc start MySQLEnterpriseMonitorAgent
```

or:

```
shell> net start MySQLEnterpriseMonitorAgent
```

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

For confirmation that the service is running you can open the Microsoft Management Console Services window. To do this go to the Control Panel, find [Administrative Tools](#) and click on the link to [Services](#). Locate the service named [MySQL Enterprise Monitor Agent](#) and look under the [STATUS](#) column.

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

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

```
shell> sc stop MySQLEnterpriseMonitorAgent
```

or:

```
shell> net stop MySQLEnterpriseMonitorAgent
```

Note

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

2.3.5.2. Starting/Stopping the Agent on Mac OS X

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

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

To stop the agent, use the [stop](#) command:

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

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

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

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

For more information about creating additional agents see, [Section 2.3.6.2, “MySQL Server \(agent-instance.ini\) Configuration”](#).

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

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

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

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

For further information on troubleshooting the agent see [Section 2.3.7, “Troubleshooting the Agent”](#).

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

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

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

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

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

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

Warning

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

2.3.5.3. Starting/Stopping the Agent on Unix

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

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

For a non-root installation the command would be:

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

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

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

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

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

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

For more information about creating additional agents see, [Section 2.3.6.2, “MySQL Server \(agent-instance.ini\) Configuration”](#).

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

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

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

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

For further information on troubleshooting the agent see [Section 2.3.7, “Troubleshooting the Agent”](#).

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

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

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

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

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

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

Warning

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

2.3.6. Advanced Agent Configuration

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

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

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

The server you want to monitor should have a directory within the specified location, optionally using the name of the server you are monitoring, and within that directory, an `agent-instance.ini` file. This file contains the configuration information for

connecting to the MySQL server, including the host name, port, user credentials and display name.

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

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

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

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

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

You can see a sample of the configuration file below:

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

[mysql-proxy]

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

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

Note

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

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

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

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

- `log-level` — sets the logging level of the agent. The default level is `message`.

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

- `debug` — provides detailed information about what the agent is doing and the information being provided by the agent to the MySQL Enterprise Service Manager.
- `critical` — lists critical messages highlighting problems with the agent.
- `error` — lists error messages.
- `warning` — provides only warning messages generated by the agent.
- `message` — provides information about the agent and basic processing information.

- `info` — provides messages used for informational purposes.

Warning

Be careful when setting the `log-level` to `debug`. Doing this will rapidly increase the size of your `mysql-monitor-agent.log` file. To avoid disk space problems, put the log files on a different drive from your MySQL server and the MySQL Enterprise Dashboard.

It is strongly recommended that you use a `log-level` of `critical` or `error` in a production server. Use the higher-levels to provide more detailed information only for debugging problems with your agent.

Under Windows, if you restart the agent from the command line after setting the `log-level` to `debug`, extensive debug information is displayed to the console as well as to the log file.

- `agent-mgmt-hostname` — sets the URL to use when reporting information. This value will be automatically set to your MySQL Enterprise Service Manager during installation.
 - `mysqld-instance-dir` — sets the directory where the configuration files that specify the MySQL servers to be monitored can be located.
 - `agent-item-files` — sets the information that is provided up to the MySQL Enterprise Service Manager when the agent is reporting status information. You should leave this item with the default setting of the `share/mysql-proxy/items/quant.lua` (which provides Query Analyzer data) and `share/mysql-proxy/items/items-mysql-monitor.xml` (which provides the core agent monitoring data).
 - `proxy-address` — sets the address and/or port number for the proxy to listen to for connections. The setting is used when employing Query Analysis as the address/port that you must configure your application to use in place of your normal MySQL server. By default this item is set during installation. The default value is 4040. If you want to support a different local host name/IP address and port, specify the host name and the port number, separated by a colon.
 - `proxy-backend-addresses` — sets the host name and port number to be used when communicating the backend MySQL server when employing query analyzer. This is the MySQL server where packets from the client are sent when communicating with the proxy on the host name/port set by the `proxy-address`.
 - `proxy-lua-script` — sets the Lua script to be used by the proxy when forwarding queries. To use Query Analyzer, this parameter should be set to `share/mysql-proxy/quant.lua`. This is the default value.
 - `agent-uuid` — sets the UUID (Universally Unique ID) of the agent. This value should be unique for all agents communicating with the same server, as the UUID is used to uniquely ID the agent within MySQL Enterprise Service Manager
- If you are setting up multiple hosts and copying the configuration between hosts, make sure that the `agent-uuid` is unique. You can have the agent create a new UUID by leaving this configuration property blank.
- `log-file` — sets the location of the log file used to record information about the agent when it is running. If you do not specify a full path name, then the log file location is considered to be relative to the installation directory of the agent.
 - `pid-file` — sets the location of the file used to record the Process ID of the agent. This is used by the script that shuts down the agent to identify the process to be shutdown. The default value is the `mysql-monitor-agent.pid` file within the base installation directory as created by the agent installer.

A number of optional parameters are also configurable within the `mysql-monitor-agent.ini` file:

- `backlog-threshold` — determines the amount of time that the agent will collect information after detecting that the service manager is down. The default value for this option is 600 seconds. In cases where there is a short network outage no information will be lost. If the outage is longer than the value of `backlog-threshold` older data is dropped as the new data is acquired.

A setting of 600 seconds means that excessive memory usage is avoided should there be a long outage. In most circumstances, there is no need to change this option. To set `backlog-threshold` to a value other than the default, add a line with the information specifying the number of seconds.

Warning

Setting this option to a value higher than the default can exhaust memory.

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

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

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

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

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

The individual configuration parameters can be defined as follows:

- `hostname` — the host name of the MySQL server that you want to monitor.
- `port` — the TCP/IP port of the MySQL server that you want to monitor.
- `user` — the user to use when connecting to the MySQL server that you want to monitor.
- `password` — the corresponding password to use when connecting to the MySQL server that you want to monitor.

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

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

2.3.6.3. Monitoring Multiple MySQL Servers

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

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

Using a Single Agent Instance

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

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

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

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

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

To add another MySQL monitored server, follow these steps:

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

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

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

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

Using Multiple Agent Instances

To use multiple agents to monitor multiple MySQL servers you need to create a new configuration structure for both the agent and the MySQL server instances you need to monitor, including the binaries and configuration files, and then update the configuration to set the corresponding parameters to monitor the new server. Using this method allows you to enable query 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 2.3.1, “Creating a MySQL User Account for the Monitor Agent”](#).
2. You need to generate a new UUID for the new agent:

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

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

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

3. Copy the main agent configuration file, `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 `mysql-d-instance-dir` to the new directory that will contain the individual MySQL server configuration files.

- Change the `proxy-address` to a different value than the first agent configuration.
 - Change the `proxy-backend-addresses` to specify the IP address and MySQL port number for the MySQL server.
 - Change the `agent-uuid` to the new value obtained in an earlier step.
 - Change the `log-file` parameter to specify a different file to use when logging errors and problems. You cannot log to the same file from two different agents.
 - Change the `pid-file` parameter to specify the file that will be used to store the process ID of the agent.
5. Copy an existing configuration directory and configuration files to the new directory:

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

6. Edit the configuration file, `instances/second/agent/agent-instance.ini` within the new directory, and set the `user`, `password` and either the `hostname` and `port`, or `socket` parameters.
7. With multiple instances, you must start each agent individually, specifying the location of the main configuration file. For example, to start the original (default) service:

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

To start the second instance:

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

2.3.6.4. Configuring an Agent to Monitor a Remote MySQL Server

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

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

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

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

Note

Monitoring a MySQL server from a remote machine affects how information is displayed in the dashboard. The `OS` and `CPU` information applies to the machine on which the agent is running not the machine hosting the monitored server. For more information on this topic see [Chapter 4, MySQL Enterprise Dashboard](#).

If your subscription level entitles you to replication autodiscovery, do **not** use remote monitoring with replication slaves or masters. The agent must be installed on the same machine as the server you are monitoring in order for discovery to work properly. For more information, see [Chapter 10, The Replication Page](#).

2.3.6.5. Monitoring Outside the Firewall with an SSH Tunnel

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

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

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

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

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

When prompted, enter the password for `user_name`.

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

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

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

2.3.6.6. Generating a new UUID

In Unix go to the command line and type:

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

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

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

This should display a line similar to the following:

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

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

Ensure that the newly created file `mysql-service-agent-3307.ini`, is in the `C:\Program Files\MySQL\Enterprise\Agent` directory.

Navigate to this directory and execute `mysql-service-agent -uf mysql-service-agent-3307.ini`. Doing this generates a new UUID and inserts it into the configuration file.

2.3.7. Troubleshooting the Agent

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

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

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

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

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

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

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

Note

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

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

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

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

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

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

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

For the location of the agent error log file see, [Section B.3, “Agent Log and PID Files”](#).

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

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

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

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

2.4. Unattended Installation

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

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

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

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

2.4.1. Command-Line Options

To view the available options for the monitor installer or for the agent installer, at the command line type the executable file name

along with the `help` option.

2.4.1.1. MySQL Enterprise Service Manager Options

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

```
--help                Display the list of valid options
--version             Display product information
--optionfile <optionfile> Installation option file
--mode <mode>        Installation mode
                    (Windows)Default: win32
                    (Unix)Default: gtk
                    (Mac OS X)Default: osx
                    (Windows)Allowed: win32 unattended
                    (Unix)Allowed: gtk text xwindow unattended
                    (Mac OS X)Allowed: osx text unattended
```

Note

The default modes are different for different operating systems. The values allowed also differ. There is no `text` installation mode under Windows.

```
--debugtrace <debugtrace> Debug filename
                    Default:
--installer-language <installer-language> Language selection
                    Default:
                    Allowed: en jp
--installdir <installdir> Installation directory
                    (Windows)Default:C:\Program »
                    Files\MySQL\Enterprise\Monitor
                    (Unix)Default:/opt/mysql/enterprise/monitor/
                    (Mac OS X)Default:/Applications/mysql/enterprise/monitor/
--tomcatport <tomcatport> Tomcat Server Port
                    Default: 18080
--tomcatshutdownport <tomcatshutdownport> Tomcat Shutdown Port
                    Default: 18005
--tomcatsslport <tomcatsslport>Tomcat SSL Port
                    Default: 18443
--usessl <usessl>    Should communication between the Dashboard »
                    and Service Manager be encrypted?
                    Default: 0
--adminuser <adminuser> Repository Username
                    Default: service_manager
--adminpassword <adminpassword>Password
                    Default:
```

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 B.5, “The config.properties File”](#).

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

The monitor installation options are the same for all operating systems except as noted in the preceding listing.

2.4.1.2. MySQL Enterprise Monitor Agent Options

To view all the options available for an unattended *agent* installation, invoke the agent installer file passing in the `help` option. (Under Windows you must redirect the output to a file as shown in [Section 2.4.1, “Command-Line Options”](#)). You should see a listing similar to the following:

```
--help                Display the list of valid options
--version             Display product information
```

```

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

```

Note

The default modes are different for different operating systems. The values allowed also differ. There is no `text` installation mode under Windows.

```

--debugtrace <debugtrace>      Debug filename
                                Default:
--installer-language <installer-language> Language selection
                                Default:
                                Allowed: en jp
--installdir <installdir>      Installation directory
                                (Windows)Default: C:\Program Files\MySQL\Enterprise\Agent
                                (Unix)Default: /opt/mysql/enterprise/agent
                                (Mac OS X)Default: /Applications/mysql/enterprise/agent
--mysqlhost <mysqlhost>        MySQL hostname or IP address
                                Default: 127.0.0.1
--checkmysqlhost <checkmysqlhost> Validation of MySQL hostname or IP address
                                Default: yes
--mysqlport <mysqlport>        MySQL port on 127.0.0.1
                                Default: 3306
--mysqluser <mysqluser>        User name on 127.0.0.1:3306
                                Default:
--mysqlpassword <mysqlpassword> Password for mysqluser on 127.0.0.1:3306
                                Default:
--managerhost <managerhost>    Hostname or IP address
                                Default: 127.0.0.1
--tomcatport <tomcatport>      Port on 127.0.0.1
                                Default: 18080
--agentuser <agentuser>        Agent username on 127.0.0.1:18080
                                Default: agent
--agentpassword <agentpassword> Agent password for agent on 127.0.0.1:18080
                                Default:
--servername <servername>     Hostname to display
                                Default:

```

Again, the agent options are the same for all operating systems except as noted.

2.4.2. Unattended Windows Installation

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

```

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

```

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

Note

Set the `installdir` and `debugtrace` options to values appropriate to your locale and operating system.

The only options that must be specified in an option file when installing the MySQL Enterprise Service Manager are `mode` (if not specified at the command line), `installdir`, and `adminpassword`.

Check the options in your option file closely before installation; no warnings will be issued if there are errors.

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

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

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

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

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

2.4.3. Unattended Unix and Mac OS X Installation

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

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

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

Note

Set the `installdir` and `debugtrace` options to values appropriate to your locale and operating system.

The only options that must be specified in an option file when installing the MySQL Enterprise Service Manager are `mode` (if not specified at the command line), `installdir`, and `adminpassword`.

Check the options in your option file closely before installation; no warnings will be issued if there are errors.

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

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

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

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

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

Note

The Service Manager does not automatically start up on rebooting. For more information, see [Bug#31676](#).

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

2.4.4. Starting the Services

For instructions on starting the services needed by the MySQL Enterprise Service Manager see, [Section 2.2.5, “Starting/Stopping](#)

the [MySQL Enterprise Monitor Service on Windows](#)” for Windows and, [Section 2.2.6, “Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X”](#) for Unix and Mac OS X.

For instructions on starting the MySQL Enterprise Monitor Agent see:

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

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

2.5. Post-Installation Considerations

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

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

For more information about the `mysql-monitor-agent.ini` file see [Section 2.3.6, “Advanced Agent Configuration”](#).

- **Configuration file** — Store the `configuration_report.txt` in a safe place. There is no mechanism for retrieving the password stored in this file.
- **Repository credentials** — The repository user name and password are stored in unencrypted form in the `config.properties` file. Take care to protect this file.
- **Disk management** — Remove installation files, and monitor the space used by the repository. Ensure that you have adequate disk space by regularly purging data. For more information, see [??? \[75\]](#)
- **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 non-standard ports, none of which are usually open by default.
- **Server upgrades** — See [Section 2.6.3.1, “Upgrading the Monitored MySQL Server”](#) for instructions on upgrading a server.
- **Repository access** — You may want to add other users.

2.6. Upgrading, Re-Installing or Changing Your Installation

You can upgrade

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

2.6.1. Upgrading MySQL Enterprise Monitor

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

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

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

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

Warning

Before updating the MySQL Enterprise Service Manager stop all agents that are reporting to that MySQL Enterprise Service Manager. If you are updating the MySQL Enterprise Monitor Agent you must also stop the MySQL Enterprise Service Manager. On a machine that runs more than one agent, the primary agent will restart when the update is complete. Any secondary agents must be restarted manually. To stop or start agents see:

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

Important

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

Warning

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

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

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

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

To update your Advisors see, [Section 2.2.7.4, “Upgrading and Updating Advisors”](#).

2.6.1.1. Upgrading from MySQL Enterprise Monitor 1.3 to 2.0

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

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

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

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

2.6.1.1.1. Upgrading to MySQL Enterprise Service Manager 2.0

Upgrading MySQL Enterprise Service Manager requires you to use one 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.
- Your 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.

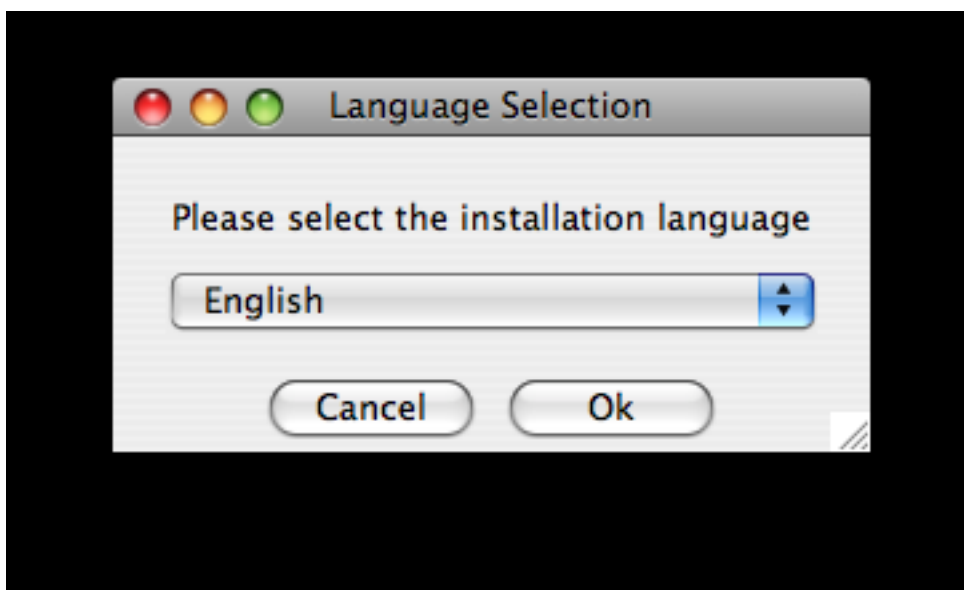
The installation of the new software using the update installer follows this basic sequence:

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

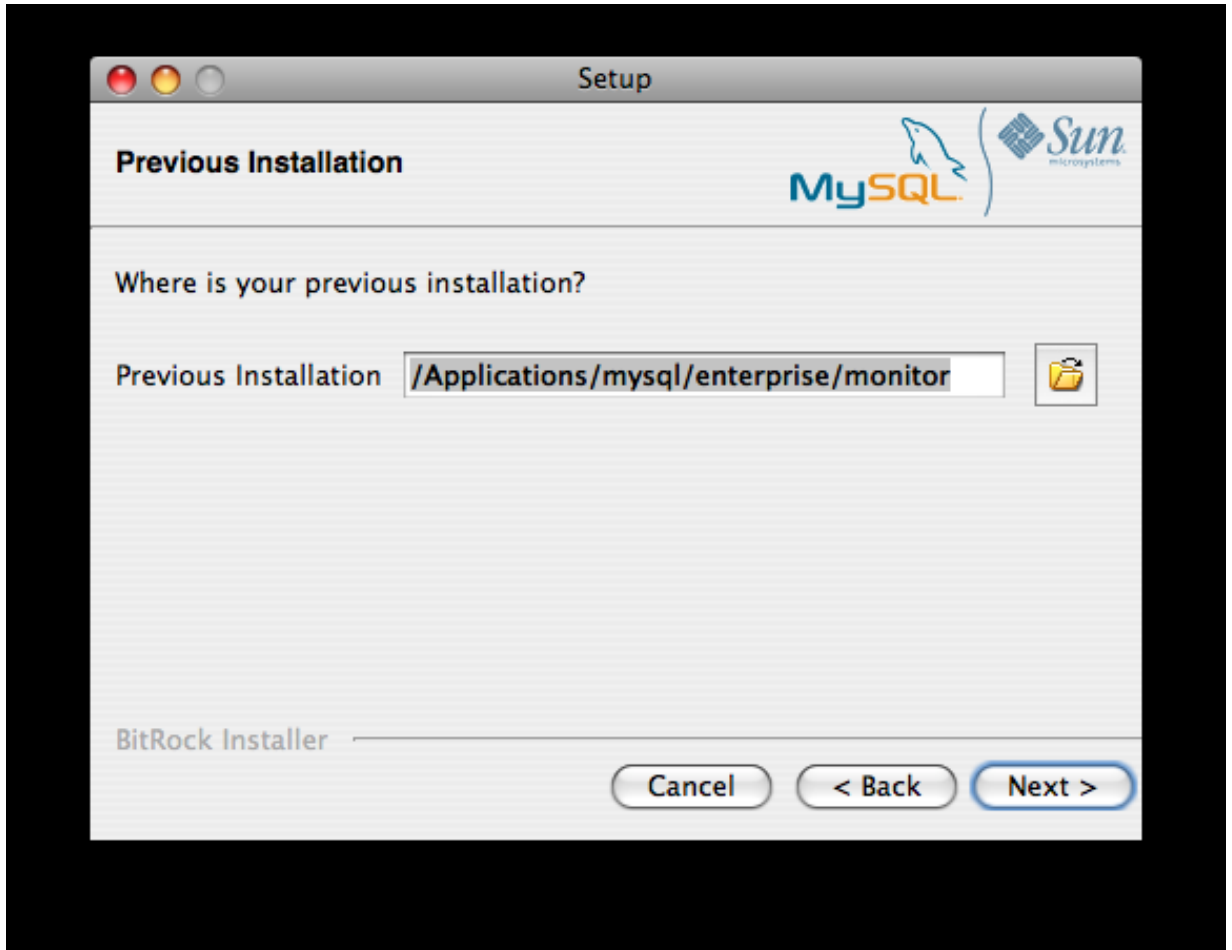
Figure 2.23. MySQL Enterprise Monitor: Server Update: Language Selection



Click OK

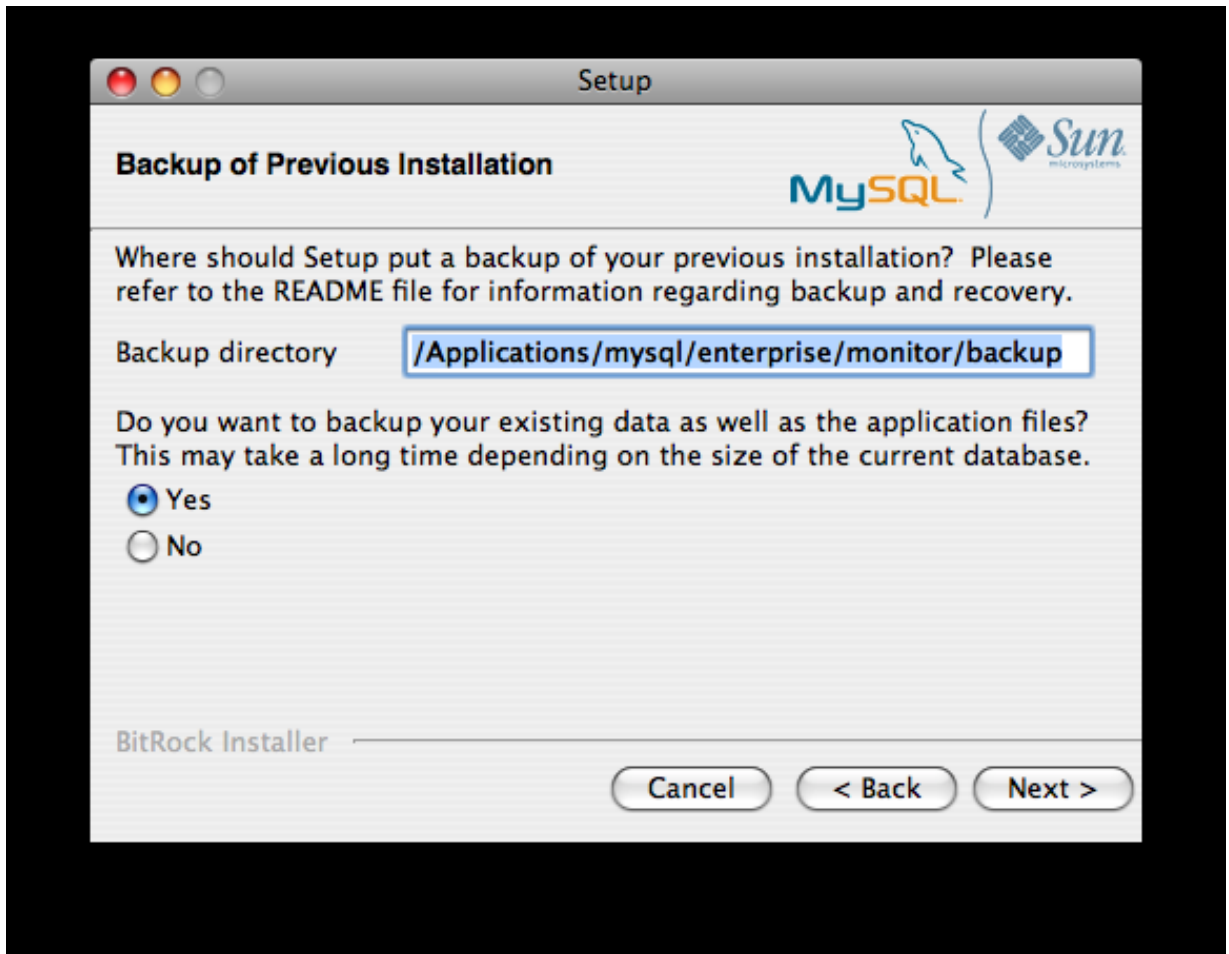
3. You will be presented with an information screen showing the application you are installing. Click NEXT to continue.
4. Specify, or locate, the previous installation of MySQL Enterprise Service Manager. If you installed the server within the default location, the current version of the application should be located automatically.

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



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

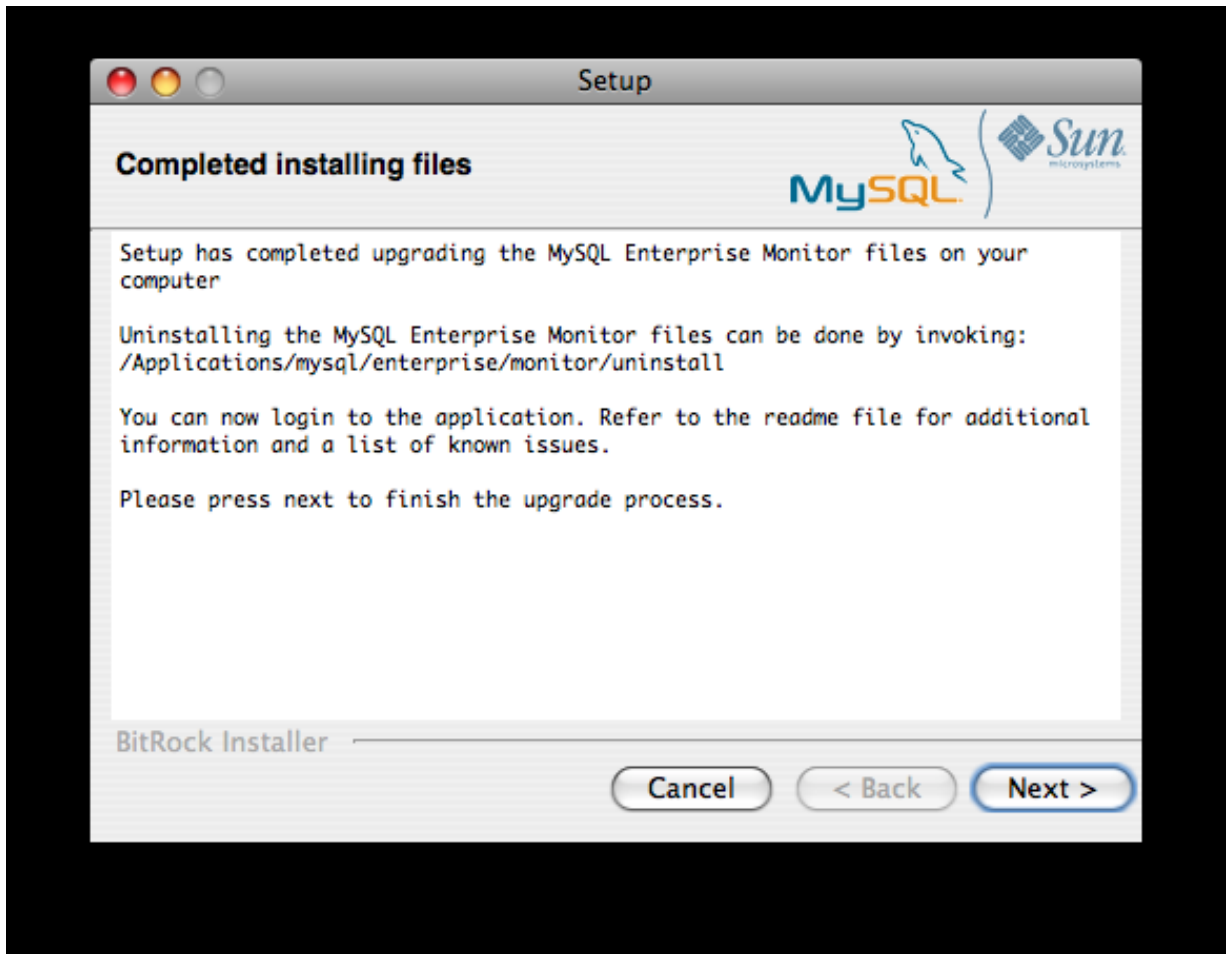
Figure 2.25. MySQL Enterprise Monitor: Server Update: Backup of Previous Installation



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

6. Specify the Tomcat Server options. The Tomcat Server Port is the default port you will use to access the MySQL Enterprise Dashboard. If you want to support agents using SSL to communicate to MySQL Enterprise Service Manager, you must check the **IS SSL SUPPORT REQUIRED?**
7. Confirm that you want to continue the installation. Once installation has started, the backup of you existing application (and database) will start, although the process may take some time. Wait until the process completes.
8. Once the process has completed you will be provided with a notification of the installation process, including how to uninstall the application if you want to do so in the future. If any errors occurred, they will be reported here.

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



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

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

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

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

2.6.1.1.2. Upgrading to MySQL Enterprise Monitor Agent 2.0

To upgrade an agent you should use a [update](#) installer. This will migrate your configuration information, simplifying the upgrade process significantly.

Note

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

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

1. Start the update installer.

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

2. Set the language for the installation process.

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

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

```
-----
Welcome to the setup wizard for the MySQL Enterprise Monitor Agent Update
-----
Please specify the directory that contains the previous installation of
the MySQL Enterprise Monitor Agent
Installation directory [/opt/mysql/enterprise/agent]:
```


- 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]:
```

- You will be asked whether you want to enable the Query Analyzer. The Query Analyzer enables you to 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 [Chapter 9, The Query Analyzer Page](#).

```
-----
Query Analysis 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-analysis.html

Enable Proxy (recommended) [Y/n]:

Proxy Port [4040]:
Backend Host: 127.0.0.1 (cannot be changed)
Backend Port: 3306 (cannot be changed)
```

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

```
-----
Setup is now ready to begin installing MySQL Enterprise Monitor Agent Update on your computer.
Do you want to continue? [Y/n]:

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

Installing
0% _____ 50% _____ 100%
#####

-----
Setup has finished installing MySQL Enterprise Monitor Agent Update on your computer.
Restart MySQL Enterprise Monitor Agent now [Y/n]:
View Readme File [Y/n]: n
```

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

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

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

Once the update agent has communicated with the MySQL Enterprise Service Manager the core information about the agent and the MySQL server it is monitoring will be migrated to the new data format required by MySQL Enterprise Service Manager 2.0. To migrate the existing stored data, see [Section 3.2, “Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0”](#).

2.6.1.2. Unattended MySQL Enterprise Monitor Update

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

```
--help                Display the list of valid options
--version             Display product information
--optionfile <optionfile>  Installation option file
                        Default:
--mode <mode>         Installation mode
                        (Windows)Default: win32
                        (Unix)Default: gtk
                        (Mac OS X)Default: osx
                        (Windows)Allowed: win32 unattended
                        (Unix)Allowed: gtk text xwindow unattended
                        (Mac OS X)Allowed: osx text unattended
--debugtrace <debugtrace>  Debug filename
                        Default:
--installer-language <installer-language>  Language selection
                        Default:
                        Allowed: en jp
--installdir <installdir>  Previous Installation
                        Default:
--createDataBackup <createDataBackup>
                        Default: 1
--backupDir <backupDir>   Backup directory
                        Default:
```

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

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

2.6.2. Reinstalling MySQL Enterprise Monitor

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

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

To stop the Monitor Agents see:

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

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

2.6.3. Changing Your MySQL Enterprise Monitor Installation

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

2.6.3.1. Upgrading the Monitored MySQL Server

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

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)

- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

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

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

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

Note

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

2.6.3.2. Changing the Server That an Agent Monitors

You need not reinstall the MySQL Enterprise Monitor Agent 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 service agent and then remove the server that it is monitoring. To stop the agent see:

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

For instructions on removing a server see, [Section 5.3.3, “Removing a Server From the Dashboard”](#).

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

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

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

To restart the agent see:

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

Note

If you are adapting an existing agent to monitor a remote server make sure that the agent has the credentials for remote access and that the port on the remote MySQL server instance is open. For more information, see [Section 2.3.6.4, “Configuring an Agent to Monitor a Remote MySQL Server”](#).

If you experience difficulties starting the agent, check [Section 2.3.7, “Troubleshooting the Agent”](#).

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

2.6.3.3. Temporarily Suspending the Agent

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

For instance, suppose you need to stop the server 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 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – see [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – see [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

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

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 6.3, “Editing Built-in Rules”](#). To blackout all events associated with a specific server or group of servers see [Section 6.6, “Advisor Blackout Periods”](#).

2.7. Uninstalling the MySQL Enterprise Monitor

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

2.7.1. Removing the MySQL Enterprise Monitor: Windows

2.7.1.1. Removing the MySQL Enterprise Service Manager

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

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

Warning

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

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

2.7.1.1.1. Removing MySQL Enterprise Monitor Services Only

When the MySQL Enterprise Service Manager is installed, the Tomcat/Apache and MySQL server services are started. It is possible to remove these services without also removing your MySQL Enterprise Service Manager installation. (For more information about these services see, [Section 2.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows”](#) or, [Section 2.2.6, “Starting/Stopping the MySQL Enterprise Monitor Service on Unix and Mac OS X”](#).)

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: `mysqlnet-workctrl help`. This batch file is discussed in more detail in [Section 2.2.5, “Starting/Stopping the MySQL Enterprise Monitor Service on Windows”](#).

2.7.1.2. 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 [Section 2.7.1.2.1, “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 2.3.6.2, “MySQL Server \(agent-instance.ini\) Configuration”](#), you will have to remove those agents manually. See the next section for instructions on doing this.

2.7.1.2.1. Removing a Single Agent

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

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

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

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

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

```
shell> sc delete AgentName
```

You can confirm that the agent has been removed by checking the Microsoft Management Console Services window. There should

no longer be an entry for the removed agent.

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

2.7.2. Removing the MySQL Enterprise Monitor: Unix and Mac OS X

2.7.2.1. 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 re-install the MySQL Enterprise Monitor.

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

Warning

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

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

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

2.7.2.2. 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 under Unix and in the `/Applications/mysql/enterprise/agent` directory on Mac OS X. 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. Under Mac OS X this directory is called `/Applications/mysql/enterprise/agent`.

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

2.7.2.2.1. 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:

- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

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

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

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

Chapter 3. Deploying MySQL Enterprise Service Manager

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 provides some notes and guidance on deploying MySQL Enterprise Service Manager, including hardware and server requirements for the MySQL Enterprise Service Manager, and how to backup the monitoring data.

3.1. 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 sample 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.

3.2. Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0

You can migrate the data generated during a MySQL Enterprise Monitor 1.3.x installation using the Data Migration functionality of the **SERVER CONFIGURATION** panel.

To use the data migration feature, you must have installed MySQL Enterprise Service Manager using an *update* installer. The update installer performs the initial migration of your configuration, rules, schedule, and events data. The historical data is not migrated until you explicitly request the migration of information within the **MANAGE SERVERS** section of the **SETTINGS** panel.

Data migration works on a single server, allowing you to select on which servers you want to migrate information. The migration is subject to the following:

- You must elect to migrate the data from each server individually.
- Migration takes approximately 5-6 hours, for each month, for each server. Therefore, if you have six months of data on 10 servers it could take between 300 and 360 hours (15 days) to migrate all of your historical data one server at a time.
- To limit the data migration, set the **DATA PURGE BEHAVIOR** within the **SETTINGS** page. Only data more recent than the specified purge period will be migrated. Data older than the purge period will be ignored.
- To prevent performance issues, migrate only one or a small number of servers concurrently.
- You can start and stop the migration of the data at any time. As a general guide, you should avoid stopping the data migration process and allow it to complete unless:
 - Run out of disk space.
 - MySQL Enterprise Service Manager becomes too slow and unresponsive.
 - Migration never completes.

With the last item, where the migration never completes, occasionally there are some aspects of the data that cannot be migrated successfully. This will prevent the migration process completing, but does not affect the conversion of any data that could be migrated.

Starting Historical Data Migration

To start data migration:

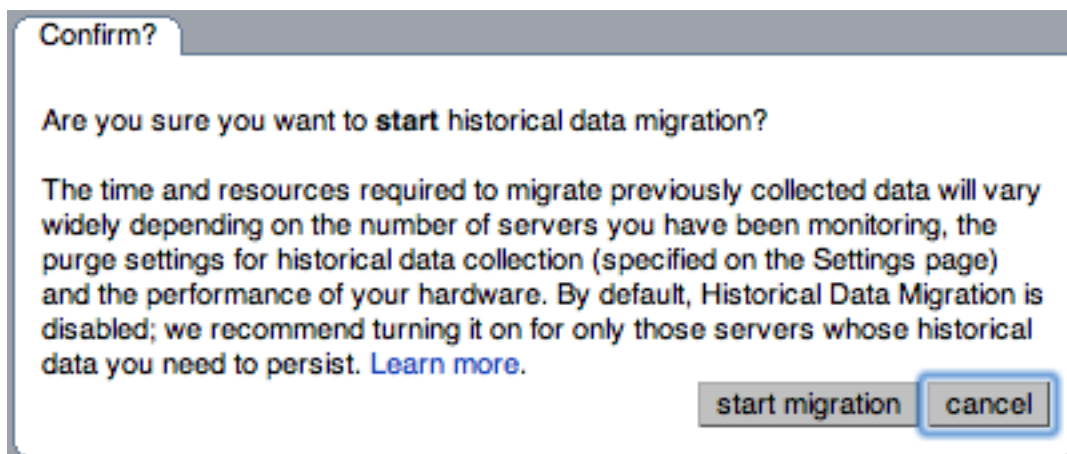
- Switch to the **MANAGE SERVERS** display of the **SETTINGS** panel within MySQL Enterprise Dashboard.
- Ensure that the data migration functionality has been enabled. The **START** and **STOP** buttons next to **HISTORICAL DATA MIGRATION** will be visible.

Figure 3.1. MySQL Enterprise Monitor: Historical Data Migration Availability

Server	Query Analyzer Enabled	Examples	Explains	MySQL	Up Since	Last MySQL	Agent	Last Agent	Migration Status
<input type="checkbox"/> All Servers (3)									
<input type="checkbox"/> bear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Sep 13, 2008 6:37:12 PM	Nov 19, 2008 1:36:00 PM	2.0.0.7085	Nov 19, 2008 1:36:19 PM	N/A
<input type="checkbox"/> gentoo1.vmbear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Nov 17, 2008 10:26:45 AM	Nov 19, 2008 2:12:00 PM	2.0.0.7101	Nov 19, 2008 1:36:19 PM	Not Migrated
<input type="checkbox"/> gentoo2.vmbear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Stopped		2.0.0.7101	Nov 19, 2008 1:36:17 PM	Not Migrated

- Select the servers you want to migrate by using the checkbox next to each server name. You can select one or more servers to migrate. Servers that are suitable for migration will show their migration status within the **MIGRATION STATUS** column. If the server is not able to be migrated, **N/A** will be shown.
- Click **START** next to **HISTORICAL DATA MIGRATION**.
- You will be presented with a confirmation dialog box. To start the migration, click **START MIGRATION**. To cancel migration, click **CANCEL**.
- The servers that have been selected for migration will show **Queued for Migration** in the **MIGRATION STATUS** column.

Figure 3.2. MySQL Enterprise Monitor: Confirming Historical Data Migration



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 3.3. MySQL Enterprise Monitor: Historical Data Migration Progress

Manage Groups & Servers									
Server	Query Analyzer			MySQL	Up Since	Last MySQL	Agent	Last Agent	Migration Status
	Enabled	Examples	Explains						
<input type="checkbox"/> All Servers (3)									
<input type="checkbox"/> bear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Sep 13, 2008 6:37:12 PM	Nov 19, 2008 1:51:00 PM	2.0.0.7085	Nov 19, 2008 1:51:55 PM	N/A
<input type="checkbox"/> gentoo1.vmbear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Nov 17, 2008 10:26:45 AM	Nov 19, 2008 2:27:00 PM	2.0.0.7101	Nov 19, 2008 1:51:55 PM	Migrating : 41 variables (50%) Start: Nov 19, 2008 1:51:28 PM
<input type="checkbox"/> gentoo2.vmbear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Stopped		2.0.0.7101	Nov 19, 2008 1:51:54 PM	Migrating : 8 variables (9%) Start: Nov 19, 2008 1:51:27 PM

Note that the migration status is shown according to the state of migration at the time the page was loaded. The actual migration continues in the background, and the current state may not match the state of the migration at the time it is viewed.

Servers showing **Done** in the **MIGRATION STATUS** column have already completed their migration.

You can check the overall migration status by examining the **UPGRADE STATUS** display.

Stopping Historical Data Migration

You can stop the migration process for any server that is still migrating data. The migration can be restarted at any time without causing any problems.

To stop the historical data migration:

1. Select the servers you want to stop migrating by using the checkbox next to each server name. You can select one or more servers to stop migrating.
2. Click **STOP** next to **HISTORICAL DATA MIGRATION**.

Confirmation that the migration has been stopped will be provided. If migration has already completed, you will be notified.

Removing Old Data

Once data migration has been completed for all the servers you want to migrate, you may want to delete or remove access to the old data within your MySQL Enterprise Monitor repository. Data for MySQL Enterprise Monitor 1.3 was stored in a database called **merlin** within the MySQL repository. Data for MySQL Enterprise Monitor 2.0 is stored within a database called **mem**.

To create a backup of the old information, use `mysqldump`:

```
shell> mysqldump -uservice_manager -pPassword -P13306 -h127.0.0.1 merlin >data-1.3.sql
```

The above will create a file, `data-1.3.sql` 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 will be removed from the **MANAGE SERVERS** panel within MySQL Enterprise Service Manager. To remove access, you need to **REVOKE** access to the `merlin` database:

```
mysql& REVOKE ALL on merlin.* FROM 'service_manager';
```

Note that revoking access to the old data will not reclaim any of the disk space used by the old data.

To delete the data from the database and free up the space being used by the historical information, **DROP** the `merlin` database:

```
mysql& DROP DATABASE merlin;
```

Once all the data has been migrated you can hide the migration user interface by clicking on the **HIDE MIGRATION INTERFACE** button.

3.3. Regular MySQL Enterprise Monitor Maintenance

MySQL Enterprise Monitor is generally self managing and does not need excessive maintenance. You should, however, be aware of certain maintenance tasks that you can automate or will need to manually perform to keep your MySQL Enterprise Monitor running efficiently.

- Make sure you have set the purge interval for your data to an appropriate value according to duration and history of data that you want to keep. For more information, see [??? \[75\]](#)
- Check, and delete, the contents of the temporary directory with your MySQL Enterprise Service Manager installation directory.

3.4. Choosing Suitable MySQL Enterprise Service Manager Hardware Configurations

Running MySQL Enterprise Service Manager places a reasonable load on your system, and this load increases linearly as you add more agents monitoring more servers. Ideally, you should use a dedicated machine for MySQL Enterprise Service Manager, rather than running it alongside other applications and services.

Minimum System Requirements

- 2 or more CPU cores
- 2 or more GB of RAM
- Disk I/O subsystem applicable for a write-intensive database

Recommended System Requirements (if monitoring 100 or more MySQL servers)

- 4 or more CPU cores
- 8 or more GB of RAM
- Disk I/O subsystem applicable for a write-intensive database (RAID10, RAID 0+1)

Chapter 4. MySQL Enterprise Dashboard

Note

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

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

Open the Dashboard by typing the host name into the address bar of your web browser. If you are unsure of the host name check the [Application host name and port](#) in the [configuration_report.txt](#) file. The default value is [http://127.0.0.1:18080/Auth.action](#) but this login is only valid if you are logging in from the machine that hosts the dashboard. If you are logging in from a remote machine you will have to specify a value other than [127.0.0.1](#). Likewise, choose a different port if you are not using the default. After logging in, select the [Monitor](#) tab.

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

From this page users can:

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

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

- [Monitor](#) — the overview page providing you with a quick summary of the servers, their status, events, availability and load. The remainder of this chapter details the contents of this page.
- [Advisors](#) — shows the various advisors configured in your installation and allows you to schedule their execution on different servers, apply and manage rules and manage the advisor installation itself. For more information, see [Chapter 6, The Advisors Page](#).
- [Events](#) — provides an interface into the event system that highlights specific issues and problems on your monitored servers. For more information on using Events, see [Chapter 7, The Events Page](#).
- [Query Analyzer](#) — interfaces to the query monitoring system that can be used to monitor and track the individual queries that are being executed on a system and help to highlight problem queries that may need optimization or that may be affecting server load. For more information, see [Chapter 9, The Query Analyzer Page](#).
- [Graphs](#) — enables you to view and configure a number of individual graphs covering a range of different statistics. For more details on how to view and use these graphs, see [Chapter 8, The Graphs Page](#).
- [Replication](#) — provides information on the status and structure of your servers that are using replication. This page is only available if you have a suitable subscription level. For more information, see [Chapter 10, The Replication Page](#).
- [Settings](#) — controls the settings for the server, including email configuration, passwords, and server and user management. For more information, see [Chapter 5, The Settings Page](#).

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

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

The [Show/Hide Legend](#) link toggles display of the key to the icons used in the [Heat Chart](#).

Note

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

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

The [Meta Info](#) section also shows how long the agent has been running, when it last contacted the MySQL server it is monitoring, and the last time the agent contacted the dashboard. Mouse over the date shown beside **UP SINCE** and a pop-up box displays the time that has elapsed since the server instance was last started. You can also mouse over the **LAST MYSQL CONTACT** and the **LAST AGENT CONTACT** dates.

Note

In the case of remote monitoring, the agent runs on a different machine than the MySQL server that it is monitoring. The [Hostname](#), [MySQL](#), and [Rules](#) information applies to the system being monitored. The [OS](#) and [CPU](#) information applies to the machine on which the agent is running. For more information about remote monitoring see, [Section 2.3.6.4, "Configuring an Agent to Monitor a Remote MySQL Server"](#).

The top of the screen shows the refresh cycle and [Help](#) and [Log Out](#) links. Click the [Help](#) link to open the documentation in a separate browser window. Choose [Log Out](#) if you wish to leave the Dashboard or to log in as a different user. Different refresh rates are available from the drop-down listbox.

In the footer are external links to MySQL Enterprise and information about the current user. Users can remain connected to the Dashboard and update their subscription, use the Enterprise Knowledge Base, and contact technical support. Your subscription information is also displayed here, showing the number of days remaining and the number of licenses. The number of licenses indicates to the number of machines that may be monitored; any number of MySQL servers may be running on a specific machine.

The footer also contains a link to the [Settings](#) page. If your subscription is current it reads **SUBSCRIPTION IS UP-TO-DATE. MORE INFO....** For more information about the [Settings](#) page see [Section 5.7, "The Product Information Screen"](#).

4.1. The Server Tree

The tab on the left displays the Server tree. By default the first group of servers is selected. This selection determines the information shown on the [Monitor](#) page.

If a server group is selected, the information presented on the [Monitor](#) page is aggregate information for this group; if only one server is selected the information applies to that server only.

Change your server selection and the information shown in the graphs and in the [Heat Chart](#) changes.

Note

For more information about server groups see, [Section 5.3.2, "Grouping Servers"](#).

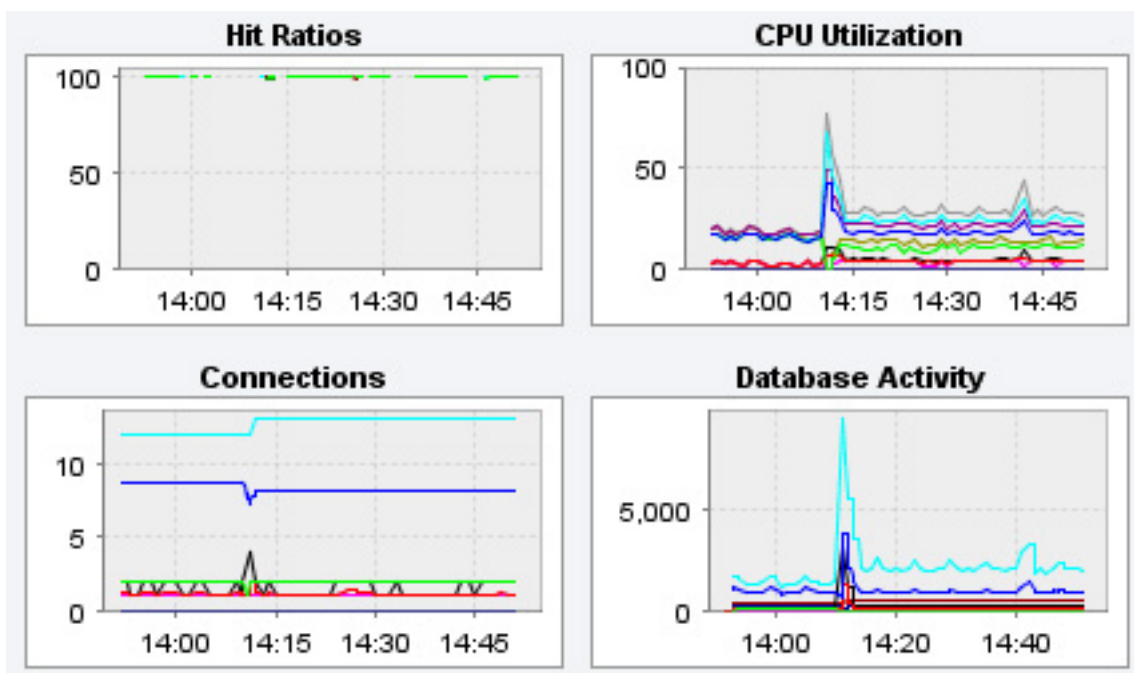
The individual server, or server group, selected in the Server Tree also determines what information appears when the [Advisors](#) tab or the [Events](#) tab is selected.

The Server Tree presents an easy way to navigate to different groups or to specific servers.

4.2. The Server Graphs and Critical Events

The center of the [Monitor](#) page gives a visual representation of the state of your servers.

Figure 4.1. MySQL Enterprise Dashboard: The Graphs screen



The graphs present information about the currently selected server or server group. The default graphs show the hit ratios, CPU utilization, connections, and database activity for a specific interval.

To set the interval click the [configure graphs](#) link immediately below the graphs. This opens a dialog box where you can choose the default interval for the x-axis of the graphs. Defining a shorter or longer interval gives you a shorter or longer term view of server activity. The thumbnail and full-size graph dimensions can also be adjusted from this dialog box. Save any changes that you have made and the values chosen will be the defaults whenever you log in.

You can also choose the default graphs shown on the [Monitor](#) page. To do this click the [edit favorites](#) link and choose the graphs you want from the drop-down list box. To choose contiguous graphs, hold down the **Shift** key and click on the desired graphs. For a non-contiguous selection, click the desired graphs while holding down the **Ctrl** key. The maximum number of graphs that can be displayed on the [Monitor](#) page is six. Save your changes and these will be the default graphs whenever you log in.

Color coding helps distinguish different aspects of each graph. With [Database Activity](#) for example, you can readily distinguish [SELECT](#) statements from database insertions.

Clicking a graph opens a detailed view with **GRAPH DISPLAY** and **CONFIGURE** tabs. Choose the [Configure](#) tab to temporarily change the way that a graph displays. Changes made from this tab only apply to the standalone graph while it is open. Persistent changes are made as described above.

Dismiss the enlarged graph by clicking the **HIDE** button.

[Critical](#) alerts appear on this page immediately below the graphs—quickly attracting your attention. For a description of all the different alarm levels see, [Section 4.3, “The Heat Chart”](#). This is the subject of discussion in [Chapter 7, *The Events Page*](#).

4.3. The Heat Chart

The [Heat Chart](#) is found on the right side of the [Monitor](#) page and shows the status of critical rules. Monitored servers are organized by groups. To view the status of a specific server, click the + button next to the appropriate server group.

Whenever a new agent contacts the Service Manager for the first time, all the rules in the Heat Chart Advisor are automatically activated. These Advisors monitor the status of the server and agent, critical operating system indicators, and important events related to your MySQL servers. An example follows.

Figure 4.2. MySQL Enterprise Dashboard: The Heat Chart

	Agent Status	Server Status	CPU Usage	IO Usage	RAM Usage	Lock Contention	My/SAM Usage	Query Cache	Temp Tables	Table Cache	Table to Disk	Critical Scans	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 4.3. MySQL Enterprise Dashboard: The Heat Chart legend

Server & Agent Status	Monitored Events
● up	○ ok
● down	● info
⊗ unknown	● warning
	● critical
	⊗ unknown

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

You may open the Heat Chart in its own browser window by clicking the [Standalone Heat Chart](#) link immediately below the [Heat Chart](#) on the left. If you like, the refresh rate can be set to a different rate than the setting on the [Monitor](#) page.

In addition to showing the most important advisors, the [Heat Chart](#) also has columns that display the number of critical, warning, and informational alarms. Clicking the hyperlink in any one of these columns takes you to the [Event](#) screen, which gives more detailed information. For more information about events see, [Chapter 7, *The Events Page*](#).

When the Dashboard is first installed no notification groups are associated with the Advisors shown in the Heat Chart. For more information on this topic see, [Section 2.2.7.3, “Installing Advisors After Initial Log-in”](#) and, [Section 5.5, “Manage Notification Groups”](#).

Chapter 5. The Settings Page

Note

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

Upon initial installation you may have configured your MySQL Enterprise credentials and also outgoing email settings. This section explores the configuration settings in more detail, and also shows how to manage servers, users, notification groups, Simple Network Management Protocol (SNMP) traps, log files, and the product information screen.

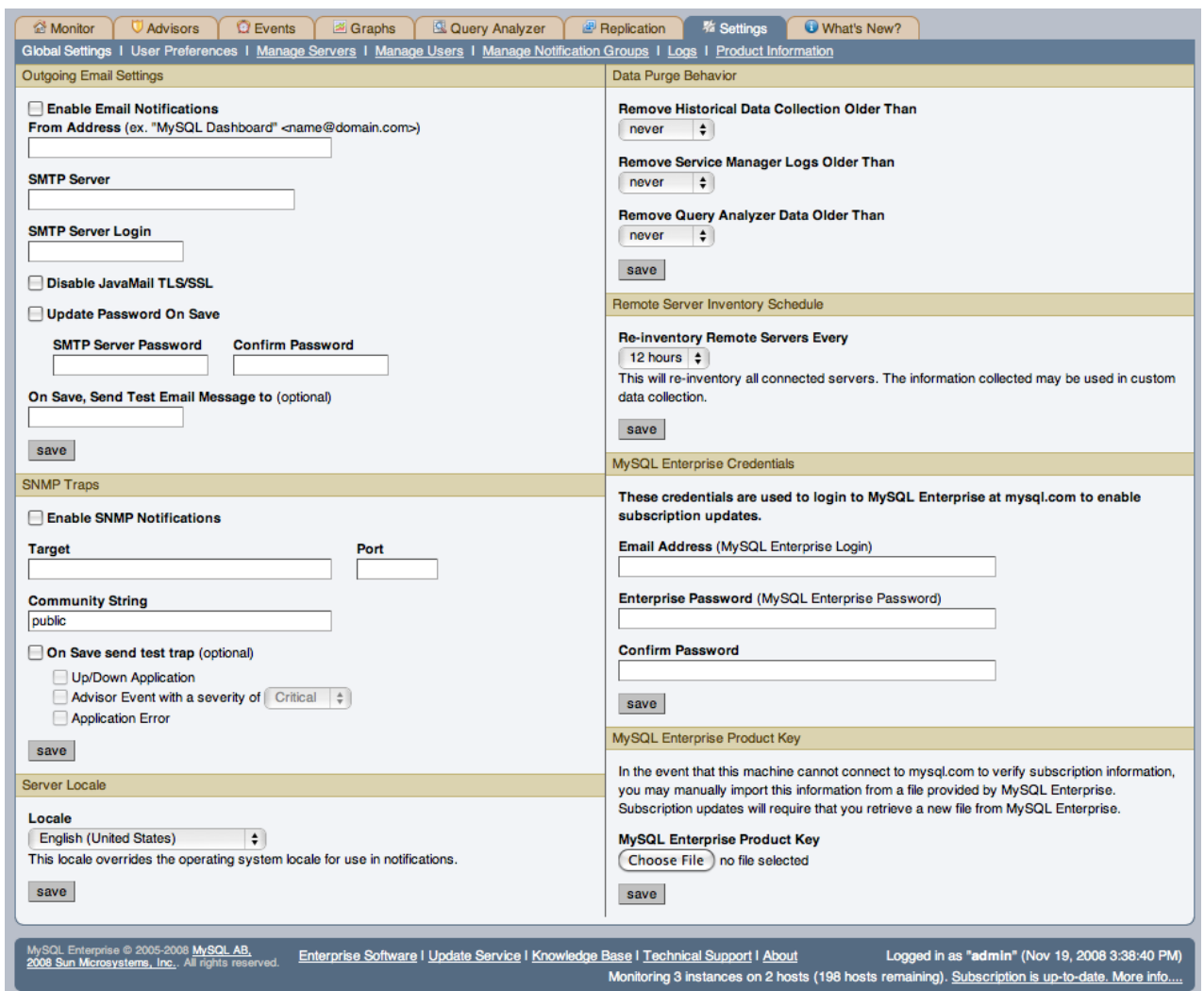
Knowledge of server management is a prerequisite for properly configuring advisors — the subject of [Chapter 6, The Advisors Page](#).

To get to the [Settings](#) page open the Dashboard and choose the [Settings](#) tab.

5.1. 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.

Figure 5.1. MySQL Enterprise Dashboard: Settings



The GLOBAL SETTINGS page is divided into a number of different sections:

- **OUTGOING EMAIL SETTINGS**

Configures the settings for email notified by MySQL Enterprise Service Manager. You must configure the **FROM ADDRESS SMTP SERVER** settings. If your server requires authorization, complete the necessary server login details, and whether SSL is required.

You can test your configuration immediately by adding an email address to the **ON SAVE, SEND TEST EMAIL MESSAGE TO** box.

For more information about [Outgoing Email Settings](#) see, [Section 2.2.7.5, “Outgoing Email Settings”](#).

- **SNMP TRAPS**

The [SNMP Traps](#) section of the [Global Preferences](#) page allows you to enable Simple Network Management Protocol so that your Network Management System (NMS) can handle events created by the MySQL Enterprise Monitor. Configure this section to route alerts and notifications to standard SNMP-enabled nodes on your network.

In the **TARGET** text box enter the IP address or the host name of your NMS listener. 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](#).

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 6.2, “Scheduling Rules”](#).

The Management Information Base (MIB) file associated with SNMP trapping is called [MONITOR.MIB](#). For the location this file see [Section B.4, “The Management Information Base \(MIB\) File”](#).

- **SERVER LOCALE**

The [Server Locale](#) setting determines the language of notification for the following items:

- Email notifications
- SNMP traps
- The naming conventions for shared resources such as a replication group name prefix

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

- **DATA PURGE BEHAVIOR**

The [Data Purge Behavior](#) section of the [Global Preferences](#) page lets you remove old log files and also old data from the repository. The default purge interval is [never](#). If you wish to purge data, change this setting by choosing from the drop-down list. Choosing [52 weeks](#), for example, will remove all data that is older than a year.

Warning

Purging data will permanently remove information from the repository. Since events are derived from data contained in the repository, they will be purged along with the data.

Ensure that there is adequate disk space for the repository. If you are monitoring numerous servers and running many rules the size of the repository can increase rapidly. Choose purge behavior accordingly.

The default value for purging, [never](#), is the safest option. However, please choose a purge setting that makes sense for your environment.

Note

The purge process is started approximately once every minute. If you change the purge duration from a larger timespan to a smaller one, the data may start to be purged immediately.

You can configure the data purge behavior for a number of different systems individually:

- **Remove Historical Data Collection Older Than** configures the duration that the main data about your servers is retained. This includes all data collections, including CPU, memory and connections and activity statistics.
- **Remove Service Manager Logs Older Than** configures the duration that the main MySQL Enterprise Service Manager logs are retained.
- **Remove Query Analyzer Data Older Than** configures the duration that the query analyzer statistics and information about individual queries is retained.

Note

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.

- **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've added or dropped databases and tables. Depending upon the configuration of your system, this operation can tax resources. If you are monitoring many remote servers this is an operation you may want to perform in off-peak hours only.

- **MYSQL ENTERPRISE CREDENTIALS**

You can specify the credentials for logging into the MySQL Enterprise Website. These should match the user name and password that you have registered with MySQL for your enterprise subscription.

Note

Only administrators can change the [MySQL Enterprise Credentials](#) section or enter a product key; for other users, this section does not show up in the interface. For more information about different users and their rights see [Section 5.4, "Managing Users"](#). Specifying incorrect credentials results in the error message, "Your credentials do not appear to be valid."

- **MYSQL ENTERPRISE PRODUCT KEY**

You may update your [MySQL Enterprise Product Key](#). If you do not have access to the Internet from the Dashboard, this provides an alternate way to update or activate the MySQL Enterprise Monitor.

To enter your product key first download it from the MySQL Enterprise website. Copy the key to a location accessible from the Dashboard. Use the BROWSE button to locate the key and then press the SAVE button.

If you wish to switch from using your MySQL Enterprise credentials to using a product key to update MySQL Enterprise Monitor, you must first clear your credentials. Do this by removing the email address from the **MYSQL ENTERPRISE CREDENTIALS** section and then clicking the SAVE button. You may then enter and save your MySQL Enterprise product key.

Note

Only administrators can change the [MySQL Enterprise Credentials](#) section or enter a product key; for other users, this section does not show up in the interface. For more information about different users and their rights see [Section 5.4, "Managing Users"](#). Specifying incorrect credentials results in the error message, "Your credentials do not appear to be valid."

5.2. User Preferences

On this page users can change their passwords, user names, and locale information.

Figure 5.2. MySQL Enterprise Dashboard: User Preferences

The screenshot shows the 'User Preferences' section of the MySQL Enterprise Service Manager interface. It includes the following fields and options:

- User Name:** admin
- Role:** manager (dropdown menu)
- Password:** (Leave empty to keep password unchanged)
- Confirm Password:**
- Timezone:** (GMT+00:00) Greenwich Mean Time - Europe/London
- Locale:** English
- Save:** A button to commit the changes.

Change your password by entering a new value into the **PASSWORD** text box. To change your user name enter a new value into the **USERNAME** text box. Click the **SAVE USER PROPERTIES** button to commit this change.

You may also adjust your time zone and locale information from this page. The settings on this page apply only to the user who is currently logged in.

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

Note

Be sure to set the correct time zone so that alerts are time stamped correctly.

This setting applies only to the specific user.

5.3. Manage Servers

To help with server management, the Service Manager supports the logical grouping of MySQL servers. This allows you to group servers in any fashion you choose. For example, you can manage servers according to purpose. You can group servers by whether the servers handle Internet or intranet data, by whether they power finance or HR applications, or, if you prefer, you may organize them by physical location rather than by functionality.

Figure 5.3. MySQL Enterprise Dashboard: Manage Servers

The screenshot shows the 'Manage Servers' panel in the MySQL Enterprise Dashboard. It includes a table of servers and several control elements:

- Buttons:** create group, configure query analyzer, Historical Data Migration: (learn more) start stop, Upgrade Status: migration is complete hide migration interface
- Table:**

Server	Query Analyzer			MySQL	Up Since	Last MySQL	Agent	Last Agent	Migration Status	
	Enabled	Examples	Explains							
<input type="checkbox"/> All Servers (3)										
<input type="checkbox"/> bear:3306 (ungrouped)	On	On	On	5.0.60-log	Sep 13, 2008 6:37:12 PM	Nov 19, 2008 4:07:00 PM	2.0.0.7085	Nov 19, 2008 4:07:13 PM	N/A	configure query analyzer rename
<input type="checkbox"/> gentool1.vmbear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Nov 17, 2008 10:26:45 AM	Nov 19, 2008 4:42:00 PM	2.0.0.7101	Nov 19, 2008 4:07:13 PM	Done	configure query analyzer rename
<input type="checkbox"/> gentool2.vmbear:3306 (ungrouped)	On	Off	Off	5.0.60-log	Nov 17, 2008 9:04:53 AM	Nov 19, 2008 1:51:00 PM	2.0.0.7101	Nov 19, 2008 4:07:11 PM	Done	configure query analyzer rename

For a server to appear in the Dashboard there must be an agent monitoring it. If you wish to add a server to the Dashboard follow the procedure for installing an agent found at [Section 2.3, "Monitor Agent Installation"](#). Instructions for adding a remote server are found at [Section 2.3.6.4, "Configuring an Agent to Monitor a Remote MySQL Server"](#).

The **MANAGE SERVERS** panel also allows you control the Query Analyzer and Data Migration. For more information, see [Section 9.6, "Query Analyzer Settings"](#) and [Section 3.2, "Migrating 1.3.x Historical Data to MySQL Enterprise Monitor 2.0"](#).

Note

The **All Servers** group is built in and every monitored server is a member of this group.

5.3.1. Renaming a Server

You can rename an existing server without losing the current historical data or configuration information. Renaming the server also allows you to modify the name of the server to be more descriptive according to the server's role within your organization. For example, you may want to rename a server from the default host name to include the department and application for the MySQL server.

To rename a server, click the RENAME link next to the server. You will be prompted with information about the server, including the host name and registered IP addresses for the agent. Fill in the alternative name that you want to be displayed in the text box at the bottom of the window.

Figure 5.4. MySQL Enterprise Dashboard: Server Renaming

Rename Server

Rename Server "bear:3306".

host
 bear.mcslp.pri
 localhost

ip
 192.168.0.2
 127.0.0.1
 192.168.92.1
 172.16.217.1

port
 3306

name

5.3.2. Grouping Servers

All monitored servers are automatically included in the top level server grouping, [All Servers](#). Other server groupings are replication groups or user-defined groups.

You can create a user-defined group by clicking on the [Manage Servers](#) link. Add a group name and then click the CREATE GROUP button. The new group will be displayed immediately.

Replication groups are automatically discovered by MySQL Enterprise Monitor and in this respect differ from user-defined groups. For more information about replication groups see [Chapter 10, The Replication Page](#). However, like user-defined groups you can edit the name of a replication group and add other servers to it.

To add to a group, select the [add to group](#) link. Choose the server or servers you wish to add and then complete the operation by choosing the ADD TO GROUP button. You can add a server to a group even if the agent is down.

To remove a server from a group expand the server group tree and click the [remove from group](#) link. To delete a server altogether see [Section 5.3.3, "Removing a Server From the Dashboard"](#).

Note

Slaves removed from a replication group will be rediscovered and re-added to that group.

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 additional 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.

5.3.3. Removing a Server From the Dashboard

If you no longer wish to monitor a MySQL server you can remove it from the Dashboard. There is no provision for deleting an active server from the Dashboard—to remove a server you must make it inactive by stopping the agent.

For instructions on stopping an agent see:

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

Once the agent is stopped you may delete the monitored server. Deleting a server simply means that it will no longer show up in the Dashboard.

Remove a server by choosing the [Settings](#) tab and then the [Manage Servers](#) link. Find the server you wish to remove and delete it by clicking the [delete](#) link. Deleting a server from the [All Servers](#) group or from any other group will remove it from the Dashboard entirely.

Note

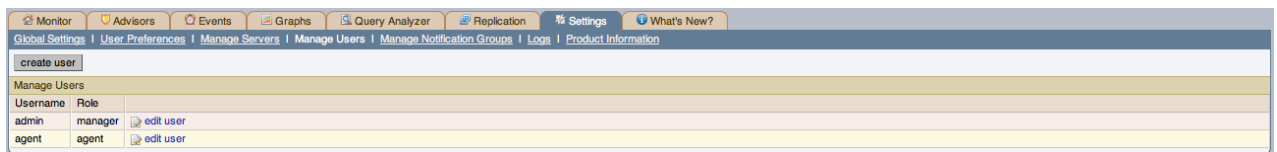
A [delete](#) link will not appear beside an active server. You must stop the agent before this link will appear.

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

5.4. Managing Users

The **MANAGE SERVERS** panel allows to create, delete and manage individual users that have access to MySQL Enterprise Service Manager

Figure 5.5. MySQL Enterprise Dashboard: Manage Users



To log in to the Dashboard a user account is required. There are three types of users with varying privileges; Administrators, Database Administrators, and Agents. The [Administrator](#) can create additional users and differs from a [DBA](#) in this respect. For this reason the [Manage Users](#) does not display if a [DBA](#) user logs in. Additionally, only administrators can change the MySQL Enterprise Credentials section or enter a product key on the [Global Settings](#) page. These sections do not appear when [DBA](#) users log in. For more information on this subject see [Section 5.1, “Global Settings”](#). The [Agent](#) account simply allows the MySQL Enterprise Monitor Agent to communicate with the Dashboard. There is no need for more than one agent account but defining an account for each server that is monitored can be an advantage since this minimizes exposure should any one agent be compromised. You cannot log in to the Dashboard using the agent’s credentials.

When the Dashboard is first launched there are two default users, [Administrator](#) and [Agent](#), both created during installation. Their default user names are respectively, [admin](#) and [agent](#). The Administrator defined during installation as having the root role is unique; this user cannot be deleted.

If you are logged in as an [Administrator](#), you can add a new user by choosing the [Manage Users](#) link from the [Settings](#) page. To create a user click the [CREATE USER](#) button, select a role for the user, and enter a user name and password.

When a new user first logs in, a dialog box opens requesting time zone and locale information. This information may be changed later from the [User Preferences](#) page. For more information, see [Section 5.2, “User Preferences”](#).

If you installed the Advisors through the Dashboard you should have already configured the settings for the root role user. (See [Section 5.1, “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 5.1, “Global Settings”](#).

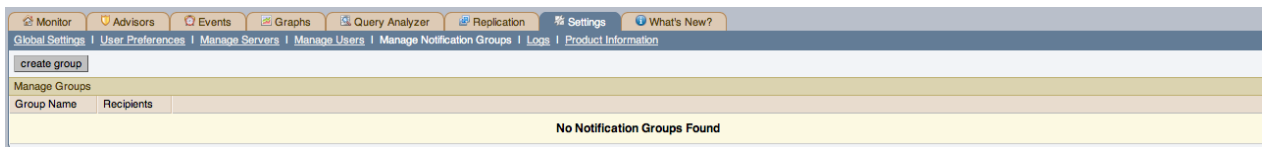
To edit an existing user’s information, select the [Manage Users](#) link, then select the user you wish to edit. Make your desired changes in the fields provided and then save your changes.

To delete an existing user, merely select the [delete](#) link.

5.5. Manage Notification Groups

The **MANAGE NOTIFICATION GROUPS** panels allows you to create and manage the notification groups used when different notifications and warnings are distributed.

Figure 5.6. MySQL Enterprise Dashboard: Manage Notification Groups



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 on 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.

If a rule triggers an alarm, an email will be sent to the members of the notification group specified when the rule was scheduled. For more information about scheduling rules see [Section 6.2, “Scheduling Rules”](#).

Note

You should ensure that there is a mail server available for sending out alerts and that there is an account configured for receiving any alerts that are created.

5.6. 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 5.7. MySQL Enterprise Dashboard: Logs

Log Name	Threshold	Last Modified	Entries	
All	N/A	Nov 19, 2008 4:07:37 PM	838	clear all logs
Advisors	Info	Nov 19, 2008 11:44:28 AM	338	clear log edit log level
Agent Monitoring	Info	N/A	0	clear log edit log level
Agent Tasks	Info	N/A	0	clear log edit log level
Data Collection	Info	N/A	0	clear log edit log level
Graphs	Info	Nov 19, 2008 11:44:12 AM	4	clear log edit log level
Groups	Info	N/A	0	clear log edit log level
Inventory	Info	Nov 19, 2008 11:44:24 AM	4	clear log edit log level
JDBC	Info	N/A	0	clear log edit log level
Misc	Info	Nov 19, 2008 11:44:29 AM	11	clear log edit log level
Migration	Info	Nov 19, 2008 1:52:09 PM	340	clear log edit log level
Notification	Info	N/A	0	clear log edit log level
Preferences	Info	N/A	0	clear log edit log level
Replication	Info	N/A	0	clear log edit log level
Security	Info	N/A	0	clear log edit log level
SQL	Info	N/A	0	clear log edit log level
Timing	Info	Nov 19, 2008 4:07:37 PM	140	clear log edit log level
Xwork	Warning	N/A	0	clear log edit log level
Apache	Warning	Nov 19, 2008 11:26:32 AM	1	clear log edit log level
Catalina	Warning	N/A	0	clear log edit log level
Hibernate	Warning	N/A	0	clear log edit log level
Freemarker	Warning	N/A	0	clear log edit log level
Spring	Warning	N/A	0	clear log edit log level
EH Cache	Warning	N/A	0	clear log edit log level

The various categories of logs are shown in alphabetical order. The most recent changes to each log are shown in the **LAST MODIFIED** column. The number of entries in any specific log is shown under the **ENTRIES** column.

To view detailed information click the [Log Name](#). This will open a separate browser window showing the date, time, alert type, and accompanying message.

On this screen you can filter log information in a couple of ways; by the message type and by time period .

To filter by message type select from the options in the **LEVEL** drop-down box. These are, in order of decreasing severity:

- All
- Error
- Warning
- Information
- Trace
- Debug

You can also adjust the number of items that appear on each page.

Press the [clear all logs](#) link to remove all log entries. To remove entries of a specific kind click the [clear logs](#) link associated with the specific log you would like to remove. A confirmation dialog box allows you to back out of this operation and avoid accidentally removing log information.

To clear log files of a specific age see the [Data Purge Behavior](#) section of the [Global Preferences](#) page. For more information on this topic see [??? \[75\]](#).

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 [??? \[75\]](#)).

Selecting [Error](#) from the list box will create the least number of log entries and [Debug](#) the most. Choosing [None](#) turns off logging altogether.

It is also possible to download a compressed version of all the log files. For more information, see [Section 5.7, “The Product Information Screen”](#).

5.7. The Product Information Screen

Use the [Product Information](#) link to view detailed information about your subscription level and 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.

Note

The **UPDATE** button was added in version 1.3 of the MySQL Enterprise Monitor. If your version of MySQL Enterprise Monitor does not have an **UPDATE** button, saving your credentials again will download a new key. See [Section 5.1, “Global Settings”](#) for instructions on doing this. If you do not have Internet access from the Dashboard, you can install a new key manually. This process is described in [Section 5.1, “Global Settings”](#).

This page also contains Enterprise Dashboard Server information; the version number, uptime, and other information related to the memory used by the Java Virtual Machine.

The [Enterprise Dashboard Server information](#) section also contains the hyperlink, [Download diagnostic report](#). Click this link to download a compressed version of the MySQL Enterprise Service Manager log files. All the log files found on the [Logs](#) page (for more information about logs see [Section 5.6, “Logs”](#)) are contained in this file. It also contains the Java properties file, the monitored MySQL servers property file, information about the status of the JDBC connection and Java threads, and the [subscription.xml](#) file. This report is especially useful for debugging the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent.

5.7.1. 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 6.1, “Installing and Updating Advisors”](#).

Chapter 6. The Advisors Page

Note

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

MySQL Enterprise Advisors are a series of scripts that gather information from your MySQL servers via the Service Manager and the Service Agents, analyze that information based on custom rules developed by MySQL AB, and then offer alerts and advice when necessary. As new rules are introduced, the MySQL Enterprise Advisors can be updated through the MySQL Enterprise website.

The MySQL Enterprise Advisors fall into the following categories:

- Administration
 - Better manage databases
 - Suggest improvements for smoother operation
- Heat Chart
 - Drive the status indicators in the Heat Chart
 - Identify up/down status and performance issues
- Performance
 - Identify potential performance bottlenecks
 - Make suggestions for improved database speed
- Replication
 - Identify replication bottlenecks
 - Improve replication design
- Schema
 - Identify schema changes
 - Find security loopholes
- Security
 - Protect MySQL servers
 - Find security loopholes

An advisor category provides a set of rules designed to enforce MySQL best practices for that specific category. Rules can be targeted to run at the individual server or group level and, upon rule violation, provide alerts and expert advice on how to address and correct a problem before it becomes a costly outage.

Individual rules are defined in the `items-mysql-monitor.xml` file. On Windows this file is found in the `C:\Program Files\MySQL\Enterprise\Agent\share\mysql-monitor-agent` directory and on Unix in the `/opt/mysql/enterprise/agent/share/mysql-monitor-agent` directory. Find below the rule for discovering a `root` account with no password.

```
<ITEM>
  <NAME>no_root_password</NAME>
  <FIELD>no_password</FIELD>
  <SCOPE>table</SCOPE>
  <CODE>
    <![CDATA[SELECT COUNT(*) AS no_password FROM mysql.user WHERE user='root' AND password='']]>
  </CODE>
  <NAMESPACE>mysql</NAMESPACE>
  <RETURNS>INTEGER</RETURNS>
  <SOURCE>table</SOURCE>
  <INSTANCE>mysql.user</INSTANCE>
</ITEM>
```

Your MySQL Enterprise subscription level determines which rules are available to you. Subscription levels are cumulative, meaning that higher MySQL Enterprise levels have access to all the rules of the lower levels.

When the Dashboard is first installed, the only rules that are scheduled are those that belong to the [Heat Chart](#) group.

Go to the Advisors screen by logging in to the Dashboard and choosing the [Advisors](#) tab.

6.1. Installing and Updating Advisors

Instructions for installing Advisors are given in [Section 2.2.7.3, “Installing Advisors After Initial Log-in”](#), and following. Principally, you need to configure your MySQL Enterprise login or enter your product key before you can update your Advisors.

If your MySQL Enterprise login is configured, you can download the latest Advisors by navigating to the [Advisors](#) page and finding the [Check for Updates](#) link. You can periodically update advisors in this way.

Note

If you do not have Internet access and cannot use the online update option you can manually import advisors. This process is described in [Section 2.2.7.3, “Installing Advisors After Initial Log-in”](#).

6.2. 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.

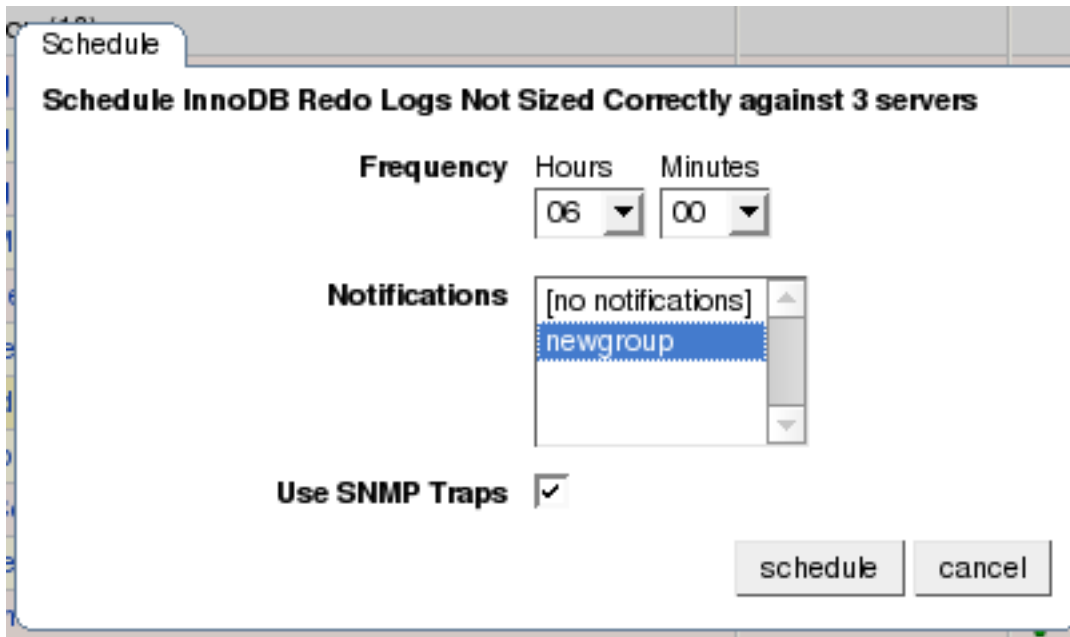
To view the advisors other than the Heat Chart group, select the [Add to Schedule](#) link. This will show all the advisors available for your subscription level.

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 6.1. MySQL Enterprise Dashboard: Scheduling Dialog



The Schedule dialog box allows you to configure the following fields:

- **Frequency** – Dictates how often the rule will run. The default value for different rules varies but a rule can be set to run at any interval desired.

Warning

Setting the frequency of a rule involves tradeoffs. Rule evaluation consumes system resources — CPU, memory, and disk space. While the amount consumed is small, if you run all the rules against dozens of servers on a very frequent basis, you may put a significant load on the Service Manager. So select an appropriate frequency. For example, unless you are stopping and restarting your servers frequently, rules that check server configuration variables probably don't need to run very often.

Another consideration is that certain status variables increase monotonically until a server is restarted. Examples of these are `Key_reads`, `Qcache_hits`, `Questions`, `Table_locks_waited`, and similar variables. The value returned by `SHOW STATUS` for these variables is the value since the server was started (or since the last `FLUSH STATUS` command), which is not very useful for performance tuning, especially if the server has been running for an extended period of time. For performance tuning it is much better to know the change in state (i.e. 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 and/or notification groups 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 5.5, “Manage Notification Groups”](#).

Set the frequency, identify whomever you wish to notify, and click SCHEDULE to schedule the advisor. Upon completion, you should see the message, [Successfully scheduled](#).

If you haven't set up global SNMP traps and would like your Network Management System (NMS) to handle events related to a specific rule then check the [Use SNMP Traps](#) checkbox. For more information about Simple Network Management Protocol (SNMP) see [Simple Network Management Protocol \(SNMP\) Traps](#).

Scheduling rules using the checkbox and the SCHEDULE button is an effective way to schedule multiple rules. To schedule a single rule you may also use the [schedule](#) link.

When scheduling more than one rule, you have the option of selecting a checkbox to use the default frequency of each rule or you may choose a frequency that will apply to all selected rules. When customizing the frequency, take care that you choose a value that is appropriate to all the rules selected.

Note

If the agent does not have the `SUPER` privilege and InnoDB-related rules are scheduled, a warning will appear in the `DataCollection` log. This also occurs if `mysqld` is started with the `skip-innodb` option. For more information about agent rights see [Section 2.3.1, “Creating a MySQL User Account for the Monitor Agent”](#).

6.2.1. 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.) Non-contiguous 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 [??? \[75\]](#).

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

6.3. 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 6.2. MySQL Enterprise Dashboard: Editing Rules

Rule Name	Advisor	Version	
Temporary Tables To Disk Ratio Excessive - copy	Heat Chart	1.0	
Expression			
(%Uptime% > 10800) && (((%Created_tmp_disk_tables% / (%Created_tmp_tables%)) * 100) > THRESHOLD)			
Thresholds			
Critical Alert	<input type="text" value="90"/>		
Warning Alert	<input type="text" value="75"/>		
Info Alert	<input type="text" value="50"/>		
Variable Assignment			
Variable	Data Item	Instance	
%Uptime%	mysql:server:Uptime	local	delete
%Created_tmp_disk_table	mysql:server:Created_tmp_disk_tables	local	delete
%Created_tmp_tables%	mysql:server:Created_tmp_tables	local	delete
%tmp_table_size%	mysql:server:tmp_table_size	local	delete
%max_heap_table_size%	mysql:server:max_heap_table_size	local	delete
<input type="button" value="add row"/>			
Default Frequency			
Hours	Minutes		
<input type="text" value="00"/>	<input type="text" value="05"/>		

Beside the rule name is the [Advisor](#) drop-down list box, used for setting the advisor group. This list box shows existing groupings and any you may have added. The [Expression](#) textarea shows the advisor rule, [Variable Assignment](#) the data item associated with variable(s) used in the rule and [Thresholds](#) determines when to trigger each alert type.

The three levels of [Thresholds](#) are [Info Alert](#), [Warning Alert](#), and [Critical Alert](#) indicating increasing levels of severity. Levels can be triggered by the expression result being equal to a certain value, greater than a certain value, or less than a certain value.

The data items that variables are associated with are operating system (OS) properties such as available RAM or MySQL characteristics such as the InnoDB buffer pool. To see all available data items drop down the [Data Item](#) list box. For a listing of these data items see [Appendix E, The Data Collection Items Used to Create Rules](#).

In [Figure 6.2, “MySQL Enterprise Dashboard: Editing Rules”](#) the drop-down [Data Item](#) list box within the [Variable Assignment](#) frame shows the various MySQL server status or operating system specific variables that may be used in expressions. The text boxes below [Thresholds](#) define the levels at which informational, warning, or critical alerts are issued.

To lower the threshold for an informational alert, simply increase the number given in the [Info Alert](#) text box.

When a data item can apply to multiple objects, you need to specify which instance to use for that item, hence the [Instance](#) text box. In almost all cases this should be set to `local`. The exceptions are as follows:

- For CPU-related items set **INSTANCE** to `cpu0`. Additional CPUs on a system are referred to as `cpu1`, `cpu2` and so on.
- There can be multiple disks mounted on a system. To refer to a specific drive set **INSTANCE** to the name of of the mounted drive. On Windows this would be `C:`, `D:`, and so on. On Unix systems, use whatever is valid for the `df` command.
- For RAM-related items set **INSTANCE** to `mem`.

- Where there are table-specific variables, the database name and table name must be specified in the **INSTANCE** text box. This topic is discussed in detail in what follows.

Note

It is not possible to have a data item that is unrelated to an instance. This raises the error, `You must map "<variable>" to an instance`, and you will be unable to save the rule.

An agent can only collect data from one MySQL server, so the `instance` entry for a variable in a rule does not need to specify which MySQL server to use; no matter how many servers are being monitored there is always a one-to-one relationship between an agent and its monitored server.

However, on one server there may be multiple occurrences of a variable. For example, there are multiple possible occurrences of table-specific variables such as `Avg_row_length` because there can be multiple databases and tables defined in a MySQL server. In this case, the “instance” refers to the database and table that a data item should be associated with, specified in the form `tablename.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.

6.4. Creating Advisors and Rules

In addition to using and editing the advisors and rules provided by MySQL Enterprise, users can create their own advisors and rules to meet their own unique needs. To do this go to the [Advisors](#) page and choose the [Create/Edit Rule](#) link.

6.4.1. Creating Advisors

Similar existing rules are grouped together in advisor groups.

The built-in advisors are:

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

The ability to create your own advisor group allows you to create groupings suitable to your circumstances.

You can create your own grouping by simply clicking the **CREATE ADVISOR** button. Enter an appropriate name and click the **ADD** button. The newly created group will appear in the [Advisor](#) column.

The newly created advisor is added to the list box of advisors shown in [Figure 6.2, “MySQL Enterprise Dashboard: Editing Rules”](#). You can now use this category of advisors when you create a new rule.

6.4.2. Overview of Rule Creation

Rules are created using the same screen seen in [Figure 6.2, “MySQL Enterprise Dashboard: Editing Rules”](#). To begin creating a rule from scratch, click the **CREATE RULE** button. However, the simplest way to create a new rule is to copy an existing one. Unlike editing an existing rule, when you copy a rule, every element of that rule is editable.

You can change the rule name, the advisor group that a rule belongs to and you can set your own version number. In [Figure 6.2, “MySQL Enterprise Dashboard: Editing Rules”](#), you have already seen how the threshold and frequency of a rule may be altered.

Most importantly you can alter a rule's expression. Expressions are the core of a MySQL Enterprise Advisor and are used to define the scenario being monitored. An expression can be as simple as a single server parameter or can be quite complex, combining multiple parameters with various mathematical operations.

An expression has two main characteristics:

- An expression defines a situation where a best practice is not being followed
- The result of an expression must always be 1 or 0 (that is, true or false)

If an expression evaluates to true for a specific server, an alarm is raised, indicating that a best practice is not being followed. If an expression evaluates to false no alarm is raised because the best practice is indeed being followed.

For example, if having binary logging enabled is considered a best practice for a production server (which we believe it is), then this best practice is being violated if `log_bin` is `OFF`. Consequently, the expression for the “Binary Logging Not Enabled” rule is `“%log_bin% == OFF”`. If this evaluates to 1, an alarm is raised because the best practice is not being followed.

An expression is made up of one or more variables and zero or more mathematical operators. The MySQL Enterprise Monitor uses the MySQL database server's expression parser and evaluator. For a complete list of operators and functions see <http://dev.mysql.com/doc/refman/5.0/en/functions.html>. For a complete list of the built-in variables used when creating 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. If you need to define more than one variable simply click the **ADD ROW** button. For a complete listing of the data collection items used in creating rules see [Appendix E, The Data Collection Items Used to Create Rules](#).

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

Note

When saving a new rule ensure that you do not duplicate the name of an existing rule.

6.4.3. Variables

When an expression is evaluated variables get replaced by values. For example, part of the expression for the “MyISAM Key Cache Has Sub-Optimal Hit Rate” rule calculates the hit rate as follows:

```
100 - ((%Key_reads% / %Key_read_requests%)*100)
```

If the current value of `%Key_reads%` is 4522 and the current value of `%Key_read_requests%` is 125989, the hit ratio assesses to 96.4%:

```
100 - ((4522 / 125989) * 100)
```

By convention, the Advisors supplied by MySQL use ‘%’ as the delimiter, for example, `%Key_reads%`. This makes variables more readily identifiable.

In addition to being used in an expression, variables may also be used in the **Description**, **Advice**, **Action**, and **Links** attributes of a rule. This allows you to report the current value of an expression.

For instance, you can add the message, “The current value of Key_reads is %Key_reads%.” to the **Advice** text box. When this is displayed on the screen, the value of `%Key_reads%` is substituted into the text. Supposing `%Key_reads%` has a value of 4522, the message becomes “The current value of Key_reads is 4522.”

6.4.4. Thresholds

Each expression has a threshold value that triggers an alert. The **THRESHOLD** keyword is used to associate that value with an alert level—either an **Info**, **Warning**, or **Critical** alert.

For example, the expression for the performance advisor, “Thread Cache Size May Not Be Optimal”, is:

```
100-((%Threads_created% / %Connections%) * 100) < THRESHOLD
```

The `THRESHOLD` is set at 95% for an Info level alert, 85% for a Warning alert, and 75% for a Critical alert; producing alerts of three different levels.

Expressions can be quite simple. The expression for “Binary Logging Not Enabled” (one of the Administration alerts) is:

```
%log_bin% == THRESHOLD
```

When the result is `OFF`, only one alert is triggered—a Warning level alert. In this situation you might think we could just use the expression `%log_bin% == "OFF"`. However, doing this would not test binary logging against a threshold so would not result in an alert.

When you create an expression, think carefully about the conditions under which it should be evaluated and the conditions under which it should not. For example, the expression for the “MyISAM Key Cache Has Sub-Optimal Hit Rate” rule is:

```
(%Uptime% > 10800) && (%Key_read_requests% > 10000) >
&& (100-((%Key_reads% / %Key_read_requests%) * 100) < THRESHOLD)
```

The essence of the rule is really: `(100-((%Key_reads% / %Key_read_requests%) * 100) < THRESHOLD)`. However, when a server is first starting up, it may take a while to reach a state that is representative of normal operations. For example, the key cache and the query cache may need some period of time before they have cached typical application data as opposed to start-up and initialization data. In this case, the first part of the expression, `(%Uptime% > 10800)`, holds off evaluating this expression until the system has been running for 10800 seconds (3 hours).

In addition, if some part of the system is not heavily used an alert may be triggered based on limited data. For example, if your application does not use the MyISAM storage engine, the “MyISAM Key Cache Has Sub-Optimal Hit Rate” rule may be triggered based on very limited use of other MyISAM tables such as the `mysql.user` table. For this reason, this advisor has a second part—`(%Key_read_requests% > 10000)`—meaning the rule won't be evaluated unless there is plenty of activity associated with the key cache.

In other circumstances, there may be periods of time during which you don't want a rule to be evaluated—a blackout period. For example, the expression for the “Slave Too Far Behind Master” rule is: `%Seconds_Behind_Master% > THRESHOLD`. However, suppose you run a backup process between 6 and 7 pm on a replication slave, and it's normal for that slave to get behind the master by an amount more than the `THRESHOLD` during that time. In that case you don't want to receive an alert because the rule violation is expected. You can achieve this by adding the following to the expression: `&& CURTIME() NOT BETWEEN '18:00:00' AND '19:00:00'` In essence, this means “don't trigger an alert between 18:00:00 and 19:00:00 (6 pm and 7 pm)”.

6.4.5. Using Strings

String values may appear in the [Expression](#) or the [Thresholds](#) text boxes. In both cases, they must be enclosed within quotation marks. For example, the expression for the “Slave I/O Thread Not Running” rule is:

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

In similar fashion the [Critical Alerts](#) threshold text box is set to a value of `"Yes"`.

When the expression is evaluated, either `"OFF"` or `"ON"` will be substituted for `%Slave_running%`, and `"Yes"` or `"No"` for `%Slave_IO_Running%`, depending on the state of your system. If the slave is running but the I/O thread is not, the expression then becomes:

```
("ON" == "ON") && ("No" != "Yes")
```

Without quotation marks this expression would not evaluate to `TRUE` as it should.

Note

So that it is interpreted properly, the `==` operator is converted to `=` before being passed to the MySQL expression parser.

6.4.6. Wiki Format

When editing or defining a rule, the text entered in the [Problem Description](#), [Advice](#), [Recommended Action](#), and [Links and Further Reading](#) text boxes may be formatted in Wiki format. This allows you to format text and add hyperlinks when creating or editing your own rules.

Find a brief introduction to using Wiki formatting in the following table.

Table 6.1. MySQL Enterprise Monitor: Wiki formatting

Example	Description
<code>__bold__</code>	boldface text
<code>~~italic~~</code>	italicize text
<code>\\</code>	create a line break
<code>\\\</code>	create a double line break
<code>\\\\G</code>	create a backslash
<code>*item 1</code>	create a bulleted list item
<code>#item 1</code>	create a numbered list item
<code>\</code>	use the ‘\’ to escape special characters
<code>{moreInfo:name url}</code>	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.

6.4.7. Creating a New Rule: An Example

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

This example creates a rule that checks the number of rows in a table. Having 50,000 rows in this table is deemed to warrant a critical alert. Lesser numbers are assigned to informational and warning level alerts.

Begin by navigating to the [Advisors](#) tab and clicking the [manage rules](#) link. Then choose the CREATE RULE button.

Create your custom rule by following these steps:

1. Using the [Rule Name](#) text box, give the rule an appropriate name. Something such as "Excessive number of records in [table_name](#) table", may be appropriate.
2. From the [Advisor](#) drop down list box choose an advisor group for your rule. The [Administration](#) group of rules might be suitable but if you wish, create your own group of advisors. For instructions on doing this see [Section 6.4.1, "Creating Advisors"](#).
3. Enter the following expression in the **EXPRESSION** text area: '%[table_name](#)_num_rows%' > THRESHOLD'. Replace [table_name](#) with the name of the table you wish to monitor. Note that the variable %[table_name](#)_num_rows% has not yet been defined.
4. Set the Thresholds.

- a. Set the `Critical Alert` level to 50000.
 - b. Set the `Warning Alert` level to 10000.
 - c. Set the `Info Alert` level to 5000.
5. Define your variable in the `Variable Assignment` frame.
 - a. In the `Variable` text box enter `'%table_name_num_rows%`, the variable used in the `Expression` text box
 - b. In the `Data Item` drop down list box find and select the `mysql:table:numrows` entry. (For a description of all the data items available see [Appendix E, The Data Collection Items Used to Create Rules](#).)
 - c. In the `Instance` text box enter `database_name.table_name`.
 6. Add appropriate entries for the `Problem Description`, the `Advice`, and the `Links` text areas. If you wish, use Wiki markup for these text areas. See [Section 6.4.6, "Wiki Format"](#) for more information. Note that you can also reference the `'%table_name_num_rows%` variable in these text areas. For example, you can display the current number of rows with a message such as `'table_name currently has %table_name_num_rows% rows.'`
 7. Save the rule.

Once the rule is created it needs to be scheduled against the server that contains the database table you wish to monitor. For instructions on scheduling rules see [Section 6.2, "Scheduling Rules"](#).

6.4.8. Creating a Custom Data Collection Item

[Section 6.4.7, "Creating a New Rule: An Example"](#) shows how to create a custom rule and [Appendix E, The Data Collection Items Used to Create Rules](#) shows the data items that can be used in rule creation. However, in some circumstances you may want to create a rule that uses a custom data collection item.

This section describes how to create a custom data collection item. The steps are as follows:

1. Create an XML file to define how the data is collected.
2. Point the agent configuration file to this XML file.
3. Restart the agent.

6.4.8.1. Creating and Using a Custom Data Item XML File

As an example, this section shows how to create a data item for monitoring the amount of free InnoDB tablespace. The format and content of the XML file that defines the data to be collected is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<classes>
  <class>
    <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>
```

Save this file as:

- Windows – `C:\Program Files\MySQL\Enterprise\Agent\share\mysql-proxy\items\innodb_min_free.xml`
- Unix – `/opt/mysql/enterprise/agent/share/mysql-proxy/items/innodb_min_free.xml`
- Mac OS X – `/Applications/mysql/enterprise/agent/share/mysql-proxy/items/innodb_min_free.xml`

After saving this file, you must point your `mysql-monitor-agent.ini` file to it. (For the location of this file on your operating system see [Section 2.3.6.1, "MySQL Enterprise Monitor Agent \(mysql-monitor-agent.ini\) Configuration"](#).) Find the `[mysql-proxy]` section and add the file name `innodb_min_free.xml` to the `item-files` parameter using a semi-colon

as a separator. For example:

```
[mysql-proxy]
...
item-files = items-mysql-monitor.xml,innodb_min_free.xml
...
```

For this change to take effect you must restart the agent. To do this see:

- Windows – [Section 2.3.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix – [Section 2.3.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X – [Section 2.3.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

Once the agent has restarted, you will find the new data item in the [Data Item](#) drop down list box on the [Rule Definition](#) page. Its fully qualified name is `mysql:table:innodb_min_free`.

6.5. Disabling and Uncheduling Rules

In some circumstances you may no longer wish to apply a rule against a specific server or group of servers and in other circumstances you may want to suspend a rule for a short length of time. With this in mind, it is possible to disable or unschedule a rule.

To disable or unschedule an advisor choose the [Current Schedule](#) screen of the [Advisors](#) tab.

Rules may be disabled or unscheduled using the buttons on the upper or lower left of the screen. You may also change a rule by clicking the [enabled](#) or [unschedule](#) hyperlink to the right of a rule. The buttons are particularly useful when you are altering more than one rule.

To no longer run a rule against a specific server, expand the advisor group and the specific rule by clicking the + button. You may then click the UNSCHEDULE button. When the dialog window opens, choose the UNSCHEDULE button and that rule will no longer be applied. If you wish to back out of the operation choose CANCEL. If, at a later date, you wish to institute this rule again, you may do so from the [Add to Schedule](#) page.

If you want to suspend a rule temporarily, use the DISABLE button and follow the same process you would for uncheduling. Once a rule is disabled the link under the status column changes to red and reads [disabled](#). When a rule is disabled, data is no longer collected for that rule. A disabled rule is easily re-enabled by clicking the [disabled](#) link or by using the ENABLE button.

Multiple rules may be altered for one or more servers by selecting the appropriate checkbox and then clicking the UNSCHEDULE, ENABLE, or DISABLE button.

Note

Rules associated with the heat chart cannot be disabled or unscheduled as they are required by MySQL Enterprise Monitor.

6.6. Advisor Blackout Periods

Database servers require regular maintenance and during these periods you may wish to stop Monitor Agents from reporting their findings. During a blackout period rules are not evaluated and notifications are put on hold but Monitor Agents continue to collect data. In this respect blacked-out rules differ from disabled rules; data continues to be collected and stored in the repository.

Blackout periods are enabled by entering the following URL into the address bar of your browser, substituting the appropriate host name, port and server name:

```
http://localhost:18080/rest?command=blackout >
&server_name=SuSE:3306&blackout_state=true
```

If you are unsure of the host name and port to use, check the [configuration_report.txt](#) file. Be sure to specify the correct port for the Tomcat server. Specify the server you wish to blackout using the name that appears in the Server Tree, being sure to include a colon and port number as shown in the preceding example.

An HTTP authentication dialog box requesting your Dashboard user name and password will open. Specify the administrator's credentials. The default user name is `admin`; use the password you specified when you initially logged in to the Dashboard.

You can also blackout a server group by entering the following URL into the address bar of your browser, substituting the appropriate host name, and server group name:

```
http://localhost:18080/rest?command=blackout >
&group_name=Finance&blackout_state=true
```

When the HTTP authentication dialog box opens, enter the administrator's credentials.

You can confirm that a server is blacked out by looking at the server name in the Dashboard; the name of a blacked out server is greyed.

To reactivate the blacked-out server or server group, use the appropriate URL and query string, changing the `blackout_state=true` name/value pair to `blackout_state=false`. Again, this must be done by a user with administrative privileges.

Note

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

6.6.1. Scripting Blackouts

Rather than opening your web browser and blacking out a server by typing entries into the address bar, you can write a script to achieve the same effect. This section documents a sample blackout script that can be run from the command line.

Create the following file and save it as `blackout.pl`.

```
#!/usr/bin/perl

use LWP 5.64;

# USAGE: blackout.pl servicemanager:18080 admin password servername:3306 true

# $ARGV[0] = management server hostname:port
# $ARGV[1] = management server username
# $ARGV[2] = management server password
# $ARGV[3] = mysqld managed instance server name and port
# $ARGV[4] = blackout state (true/false)

my $browser = LWP::UserAgent->new;
$browser->credentials(
    $ARGV[0],
    ' ',
    $ARGV[1],
    $ARGV[2]
);

my $url = URI->new('http://'.$ARGV[0].'/rest');

$url->query_form( # And here the form data pairs:
    'command' => 'blackout',
    'server_name' => $ARGV[3],
    'blackout_state' => $ARGV[4]
);

my $response = $browser->post( $url );

if (!$response->is_success) {
    die $response->status_line . "\n";
}
```

Note

Windows users can omit the shebang line.

On Unix systems use the `chmod +x blackout.pl` command to make the file executable.

At the command line enter `blackout.pl servicemanager:18080 admin password servername:3306 true`.

If you are unsure of the host name and port to use, check the `configuration_report.txt` file. Be sure to specify the correct port for the Tomcat server. Specify the server you wish to blackout using the name that appears in the Server Tree, being sure to include a colon and port number as shown in the preceding example. Make sure that the user you specify is a "manager". Specifying a user with "dba" rights only will not black out a server and no error will be displayed.

You can confirm that a server is blacked out by looking at the server name in the Dashboard; the name of a blacked out server is greyed. To end the blackout, run the same script, changing the final argument to `false`.

Note

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

Chapter 7. The Events Page

Note

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

Once an advisor has been scheduled, it will run at set intervals. If it finds nothing of interest no alerts or emails will be created.

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 5.5, “Manage Notification Groups”](#).

To view open events, click on 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 7.1. MySQL Enterprise Dashboard: Events screen

Severity	From	To	Advisors	Rules
All			All Categories	All

Status	Limit	filter	reset
All	20		

Severity	Server	Advisor	Rule	Time	Status	
	susieQ:3306	Heat Chart	CPU I/O Usage Excessive	2/22/2007 12:51 PM	Open	
	susieQ:3306	Heat Chart	CPU Usage Excessive	2/22/2007 12:35 PM	Open	
	susieQ:3306	Heat Chart	RAM Usage Excessive	2/22/2007 11:35 AM	Open	
	susieQ:3306	Heat Chart	MySQL Agent Not Reachable	2/22/2007 7:01 AM	Open	
	susieQ:3306	Heat Chart	Table Scans Excessive	2/21/2007 8:24 PM	Open	
	susieQ:3306	Security	Users Can View All Databases On MySQL Server	2/21/2007 8:20 PM	Open	
	susieQ:3306	Security	Server Includes A Root User Account	2/21/2007 8:20 PM	Open	
	susieQ:3306	Security	Server Has Anonymous Accounts	2/21/2007 8:20 PM	Open	

The event table has the following columns:

- **SEVERITY** – An icon indicating the severity of the alert
- **SERVER** – The name of the server the alert applies to
- **ADVISOR** – The category of the advisor
- **RULE** – A short description of the rule that has been violated
- **TIME** – The approximate time the event occurred
- **STATUS** – The status of the event
- **UNNAMED COLUMN** – Provides a link to the **Close** dialog box

By default, all events are shown but the list of events can be filtered using the form displayed above the event list. The options include filtering by:

- Severity
- Date
- Advisor group
- Specific rule
- Status

Choose the options you are interested in and click the **FILTER** button to refresh the display. You may limit the number of items that appear on a page by choosing a different value from the **LIMIT** drop down listbox.

The drop down list box showing severity has the options: [All](#), [Alerts](#), [Critical](#), [Warning](#), [Info](#), [Success](#), and [Unknown](#). Selecting the option [All](#) shows all alerts and also those rules that have run successfully. A successful rule is one that has not been violated and is indicated by a star icon. [Alerts](#) shows only those rules that have been violated.

Columns are sorted by clicking on the individual column headings. The alerts shown in [Figure 7.1, “MySQL Enterprise Dashboard: Events screen”](#), are sorted by decreasing severity. An octagonal red icon indicates a critical alert, a triangular yellow icon a warning, and a conversation bubble an informational alert. A star beside an event indicates that the rule has run successfully and no alert created. A question mark icon indicates that the status of the rule is unknown.

The server shown in [Figure 7.1, “MySQL Enterprise Dashboard: Events screen”](#), is filtered for [All](#). Typically, when filtering by severity you would choose [Alerts](#) and, if you see a [Critical](#), [Warning](#), or [Info](#) alert, use the [All](#) filter to see when the rule last ran successfully. This may assist in determining the cause of the alert.

Besides filtering for severity, you can also choose to filter for a specific time period using the [From](#) and [To](#) text boxes. You also have the choice of filtering by specific rules or categories of rules. The [Status](#) drop-down list box let's you choose [All](#), [Open](#), or [Closed](#) events. To avoid excessive scrolling, you can also limit the number of events that show on a specific page.

For more information about an alert, click on 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.

7.1. Closing an Event

After determining what action to take, events should be closed.

To resolve an individual alert click the [close](#) link in the **OPERATIONS/NOTES** column. Document the resolution using the [Notes](#) text area and choose the **CLOSE EVENT(S)** button.

To close a number of alerts simultaneously, select the checkbox beside the alerts you wish to close and then click the **CLOSE** button to the lower or upper left side of the screen.

Once an event has been closed it appears on the [Events](#) screen showing a [resolution notes](#) link. Click this link to review the notes. Events that have been closed are saved in the Repository. If you wish to view closed events filter the display by choosing [Closed](#) from the **STATUS** drop-down box.

Chapter 8. The Graphs Page

Note

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

Navigate to the [Graphs](#) page by choosing the **GRAPHS** tab.

By default four graphs are displayed on the [Monitor](#) page. These graphs present information about the currently selected server or server group, showing the hit ratios, CPU utilization, connections, and database activity. Color coding helps distinguish different aspects of each graph.

From the [Monitor](#) page you can make permanent or temporary changes to the way a graph is displayed. For example, you can choose to display the last hour's activity or you can choose to view a specific period of time.

Persistent changes to the way the graphs display are only made from the [Monitor](#) page. You can set the size of the thumbnails and the full-sized graphs and you can also set their refresh interval. For more information, see [Section 4.2, "The Server Graphs and Critical Events"](#). As with the [Monitor](#) page, the data shown in the graphs is determined by the server or group of servers selected in the server tree.

The [Graphs](#) page shows all the available graphs and provides the capability of adjusting the scale of the graphs, allowing a more or less detailed view as the situation requires. To ensure that you have the latest versions of the various graphs click on the **CHECK FOR UPDATES** link on the top left of this page.

8.1. Displaying Graphs

The total number of graphs varies depending upon your subscription level. The four graphs that appear by default on the [Monitor](#) page are:

- Hit Ratios
- Database Activity
- Connections
- CPU Utilization

When the [Graphs](#) page is first opened, no graphs are visible. To view a graph click the + button on the left or, to view all graphs, use the **EXPAND ALL** button.

The larger size of graphs is the primary reason for viewing graphs on the [Graphs](#) page rather than on the [Monitor](#) page. Additionally, you can only show a maximum of six graphs on the [Monitor](#) page; the remaining graphs can only be viewed from the [Graphs](#) page.

8.2. Setting an Interval

Change the interval for a graph by choosing values from the **HOURS** and **MINUTES** drop-down list boxes. If necessary adjust the width and height of the graph and then click the **UPDATE** button. The changes to the time span apply to all the graphs on the [Graphs](#) page but have *no* effect on the graphs on the [Monitor](#) page.

To change the graphs both here and on the [Monitor](#) page use the [configure graphs](#) link on the top right. This opens a dialog box for setting the default interval for the x-axis. Save any changes that you have made and the values chosen will be the defaults whenever you log in. You can also change the defaults from the [Monitor](#) page as described in [Section 4.2, "The Server Graphs and Critical Events"](#); defaults for other users will be unchanged.

Use the **RESET** button to restore the default value for the interval. Doing this will also reset the default size of the graphs.

8.3. 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](#) page. You cannot change the time span of the graphs that appear on the [Monitor](#) page. Changes apply only to the current user; defaults for other users will be unchanged.

Use the **RESET** button to cancel your changes.

Chapter 9. The Query Analyzer Page

Note

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

Query Analyzer enables you to monitor the statements being executed on a monitored server and retrieve information about the query, number of executions and the execution times of each query. Queries are normalized, so that the unique data defined within each query has been removed. By removing the data specific elements of the queries, the generic queries can be counted and identified more easily.

Important

After this release is pronounced Generally Available, the MySQL Agent will be ready for use in your production systems; however, we strongly recommend reading these guidelines before using/deploying MySQL Query Analyzer.

MySQL Query Analyzer is designed to gather query performance information from a variety of sources. In this initial release, 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 `EXPLAINS` 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 initial 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. We recommend carefully monitoring 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 non-peak hours or monitoring only a subset of queries on this system.

If you experience any problems with Query Analyzer, we're interested in working with you closely and quickly to resolve them. Please open a Support issue right away. We're already working hard on optimizing Agent/proxy RAM usage and are planning a series of rapid releases to quickly distribute these and other improvements to you.

Query Analyzer works by intercepting the SQL statements that your MySQL client application sends to the MySQL server. Instead of connecting direct to the MySQL Server, queries are routed through the MySQL Enterprise Monitor Agent, the agent/proxy forwards the queries on to the server and sends the replies back to the client application as normal. In addition to forwarding the queries, the agent/proxy will also normalize the queries and then supply the execution information about each query to the monitor. The forwarding functionality is provided by the same module that supports the MySQL Proxy application. For information on MySQL Proxy, see [MySQL Proxy](#).

Important

The MySQL Proxy component, and Query Analyzer, require that clients connecting through MySQL Enterprise Monitor Agent are using MySQL 5.0 or later. Clients that use the library provided with MySQL 4.1 or earlier will not work with MySQL Enterprise Monitor Agent.

Once your MySQL client application has been configured to communicate via the MySQL Enterprise Monitor Agent, queries will be monitored and the simplified queries, without the query specific data, will be sent to the MySQL Enterprise Monitor Agent.

There are a number of different ways that you can enable Query Analysis. For more information on the different options, see [Section 9.1, "Enabling Query Analyzer"](#).

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

Figure 9.1. MySQL Enterprise Dashboard: Query Analyzer

Query	Database	Exec Count	Exec Time (hh:mm:ss.ms)			Rows			Bytes			First Seen
			Total	Max	Avg	Total	Max	Avg	Total	Max	Avg	
SELECT COUNT(message_i... process_type , fmdtdate (1)	intranet_mcslp	6	12.838	11.123	2.140	776	408	129	24.21 KB	12.8 KB	4.04 KB	10:35:01 AM
SELECT inhost , path , ...GROUP BY fmdtdate , path (1)	intranet_mcslp	2	0.906	0.686	0.453	4,038	2,526	2,019	230.89 KB	142.05 KB	115.44 KB	10:34:01 AM
SELECT process_mode , p...ess_mode , process_type (1)	intranet_mcslp	6	1.674	0.333	0.279	36	7	6	808 B	165 B	134.67 B	10:37:01 AM
SELECT media_photo . ph...RDER BY RAND() LIMIT ? (1)	intranet_mcslp	1	0.259	0.259	0.259	30	30	30	282 B	282 B	282 B	10:37:01 AM
SELECT process_mode , p...ess_mode , process_type (1)	intranet_mcslp	1	0.241	0.241	0.241	8	8	8	189 B	189 B	189 B	10:33:01 AM
SELECT data , COUNT(DI...P BY data ORDER BY data (1)	intranet_mcslp	1	0.217	0.217	0.217	195	195	195	3.06 KB	3.06 KB	3.06 KB	10:39:01 AM
SELECT DISTINCT(media... , photoid DESC LIMIT ? (1)	intranet_mcslp	25	1.781	0.209	0.071	250	10	10	2.44 KB	100 B	100 B	10:38:01 AM
SELECT DISTINCT(album...RDER BY RAND() LIMIT ? (1)	intranet_mcslp	1	0.193	0.193	0.193	10	10	10	594 B	594 B	594 B	10:37:01 AM
INSERT INTO currencies ...ALUES (? , ? , ? , ?) (1)	intranet_mcslp	5	0.109	0.105	0.022	5	1	1	0 B	0 B	0 B	10:31:01 AM
SELECT COUNT(DISTINCT(...oto_meta WHERE type = ? (1)	intranet_mcslp	1	0.095	0.095	0.095	1	1	1	8 B	8 B	8 B	10:37:01 AM
SELECT COUNT(videoid...?) AND last_view > ? (1)	intranet_mcslp	5	0.100	0.091	0.020	5	1	1	35 B	7 B	7 B	10:37:01 AM
INSERT INTO stocks (da...ALUES (? , ? , ? , ?) (1)	intranet_mcslp	2	0.078	0.071	0.039	2	1	1	0 B	0 B	0 B	10:31:01 AM
SELECT * FROM statmon...e_disk WHERE statid = ? (1)	intranet_mcslp	92	3.202	0.066	0.035	345	9	4	15.25 KB	431 B	169.75 B	10:31:01 AM
SELECT SUM(data) , CO... media_video . type = ? (1)	intranet_mcslp	2	0.082	0.064	0.041	2	1	1	33 B	17 B	16.5 B	10:37:01 AM
SELECT media_photo . ph...RDER BY RAND() LIMIT ? (1)	intranet_mcslp	24	0.231	0.061	0.010	1,206	150	50	10.91 KB	1.37 KB	465.58 B	10:38:01 AM
SELECT DISTINCT(machin...DATE() , logtime) < ? (1)	intranet_mcslp	23	0.786	0.055	0.034	92	4	4	575 B	25 B	25 B	10:31:01 AM
INSERT INTO markets (d...ALUES (? , ? , ? , ?) (1)	intranet_mcslp	7	0.057	0.047	0.008	7	1	1	0 B	0 B	0 B	10:31:01 AM
SELECT DISTINCT(media...RDER BY RAND() LIMIT ? (1)	intranet_mcslp	1	0.042	0.042	0.042	60	60	60	2.29 KB	2.29 KB	2.29 KB	10:37:01 AM
SELECT COUNT(DISTINCT(...st)) FROM media_audio (1)	intranet_mcslp	1	0.033	0.033	0.033	1	1	1	9 B	9 B	9 B	10:37:01 AM
SELECT (COUNT(DISTINC... * ?) FROM media_audio (1)	intranet_mcslp	2	0.065	0.033	0.032	2	1	1	25 B	13 B	12.5 B	10:37:01 AM

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 9.3, “Filtering Query Analyzer Data”](#).

Each row within the table provides the statistical information for one normalized query statement. If you have configured multiple agent/proxies to accept and forward queries to different servers, then you can expand the server view. The summary information displayed is different depending on whether you have selected a server group or an individual server.

If you have selected a server group, then the information displayed is aggregated from across the entire group. The same query executed on multiple servers will show average, total and minimum/maximum information for that query across all the servers. If you select an individual server, then only queries executed on that server are included within the table.

For each row, the following columns are populated according to the selected filtering options. For example, if the filter have been configured to show queries within the last 30 minutes (**INTERVAL**), then only queries executed during that time will be displayed, and the corresponding statistics, such as execution times, rows returned and bytes returned will be according to that 30 minute timespan.

- **QUERY** — the normalized version of the query. Normalization removes the query-specific data so that different queries with different data parameters are identified as the same basic query.

The information is shown as one query per row. Each query row is expandable, and can be expanded to show the execution times for individual servers for that query.

- **DATABASE** — the name of the database used in the query. The column may be blank if the database name has not been explicitly stated within the query.
- **EXEC COUNT** — the number of times that the query has been executed.
- **EXEC TIME** — the execution time for all the matching queries. This is the time, for every invocation of the corresponding query, as calculated by comparing the time when the query was submitted and when the results were returned by the server. Times are expressed in HH:MM::SS.MS (hours, minutes, seconds, and milliseconds).

The **EXECUTION** column is further subdivided into the following columns:

- **COUNT** — the total number of executions.
- **TOTAL** — the cumulative execution time for all the executions of this query.

- **MAX** — the maximum execution time for an execution of this query.
- **AVG** — the average execution time for the execution of this query.

When looking at the information provided in this query, you should consider comparing the average and maximum execution times to see if there was a problem on a specific server or during a specific time period when the query took place, as this may indicate an issue that needs to be investigated. For more information, see [Section 9.4, “Using Query Analyzer Data”](#).

- **ROWS** — the rows returned by the query. The column is sub-divided into the following columns:
 - **TOTAL** — the sum total number of rows returned by all executions of the query.
 - **MAX** — the maximum number of rows returned by a single execution of the query.
 - **AVG** — the average number of rows returned by all executions of the query.
- **BYTES** — the number of bytes returned by each query. The column is sub-divided into the following columns:
 - **TOTAL** — the sum total bytes returned by all executions of the query.
 - **MAX** — the maximum number of bytes returned by a single execution of the query.
 - **AVG** — the average number of bytes returned by all executions of the query.
- **FIRST SEEN** — the first time the query was seen within the given filter conditions.

You can sort the list of queries by clicking on 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.

9.1. Enabling Query Analyzer

There are three different ways of enabling query analyzer:

- Change your MySQL client application to talk to the Proxy port you configured during installation. This requires changing your MySQL client application code, and may require that you stop and restart your MySQL client application, but does not require any changes to your MySQL server. For more information, see [Section 9.1.1, “Enabling Query Analyzer by Changing the MySQL Client Application”](#).
- Change your MySQL server to listen on a different port, and configure the Agent/proxy to listen on the original MySQL server port. This does not require any changes to your MySQL client application, but will require shutting down and restarting your MySQL server, which may affect your cache and performance. For more information, see [Section 9.1.2, “Enabling Query Analyzer by Changing MySQL Server”](#).
- Use IP tables to redirect the network packets to the agent/proxy.

Important

After this release is pronounced Generally Available, the MySQL Agent/proxy will be ready for use in your production systems; however, we strongly recommend reading these guidelines before using/deploying MySQL Query Analyzer.

MySQL Query Analyzer is designed to gather query performance information from a variety of sources. In this initial release, 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 [EXPLAINS](#) 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 initial 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. We recommend carefully monitoring 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 non-peak hours or monitoring only a subset of queries on this system.

If you experience any problems with Query Analyzer, we're interested in working with you closely and quickly to resolve them. Please open a Support issue right away. We're already working hard on optimizing Agent/proxy RAM usage and are planning a series of rapid releases to quickly distribute these and other improvements to you.

Note that you must have enabled Query Analyzer within the agent/proxy during installation. If you did not enable Query Analyzer during the installation of the agent/proxy, check the following elements within the main `mysql-monitor-agent.ini` configuration file:

- Add the `proxy` plugin to the `plugins` parameter:

```
plugins=proxy,agent
```

- Ensure that the `quan.lua` items file is enabled in the `agent-item-files` configuration property:

```
agent-item-files = share/mysql-proxy/items/quan.lua,share/mysql-proxy/items/items-mysql-monitor.xml
```

- Check and set the `proxy-address`, `proxy-backend-addresses`, and `proxy-lua-script` settings are configured:

```
proxy-address=:4040
proxy-backend-addresses = 127.0.0.1:3306
proxy-lua-script = share/mysql-proxy/quan.lua
```

For more information on these configuration options, see [Section 2.3.6.1, "MySQL Enterprise Monitor Agent \(mysql-monitor-agent.ini\) Configuration"](#).

Note

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

You may also need to make some additional changes to the security configuration on your server to ensure that queries are correctly reported to MySQL Enterprise Service Manager:

- You must ensure that each user configured within your MySQL client application that connects through the agent/proxy and is required to report query analyzer information is allowed to connect to the server from the host on which the agent/proxy is running. When the user connects to the agent/proxy, and the agent/proxy connects to the server the host of the agent/proxy will be used as the identifying client host name during the connection.

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:

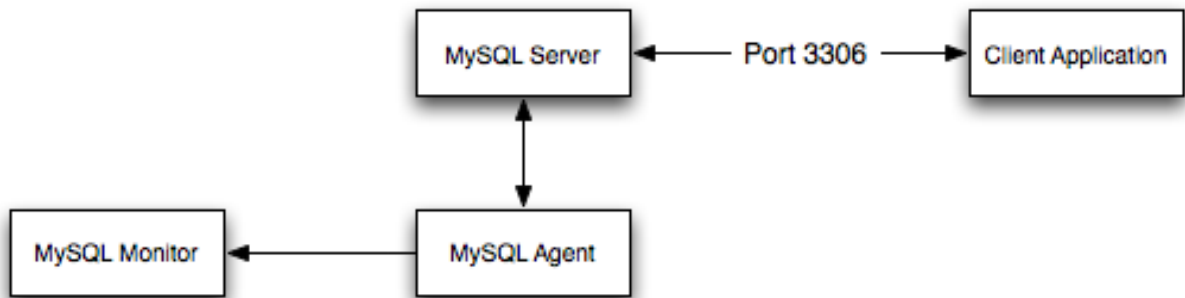
```
mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
```

9.1.1. Enabling Query Analyzer by Changing the MySQL Client Application

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

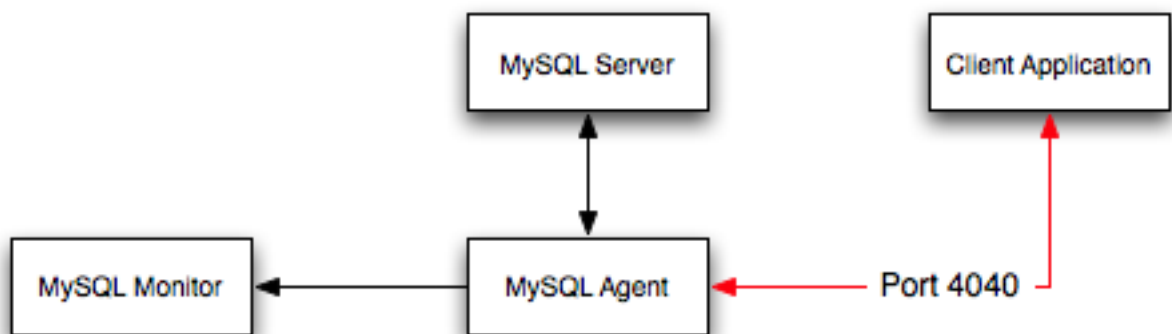
with the MySQL server, but to the agent/proxy.

Figure 9.2. MySQL Enterprise Dashboard: Standard agent/monitor topology



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

Figure 9.3. MySQL Enterprise Dashboard: Query Analyzer agent/monitor topology



To enable query analyzer within your MySQL client application:

1. Make sure that the MySQL Enterprise Service Manager and your MySQL Enterprise Monitor Agent are configured and running.
2. Confirm the configuration of your agent by examining the contents of the `etc/mysql-monitor-agent.ini` file within your installed Agent directory.

Queries will be sent to the host specified in the `proxy-backend-addresses` parameter, and the agent will listen for connections to be redirected to the server on the host name and port configured in the `proxy-address` parameter.

3. Now modify your MySQL client application to communicate with the address specified in the `proxy-address` parameter.

Alternatively, if you do not want to modify your application directly, you can use iptables or firewall rules to redirect queries from the original host/port combination to the agent's port.

Because connections to the MySQL server will be coming from the agent/proxy, not the original host, the user credentials used must 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=4040 --protocol=tcp
shell> mysql --port=4040 --host=127.0.0.1
```

Note

It is recommended that you use one agent/proxy per MySQL server instance. The agent/proxy is not able to forward queries to multiple MySQL server backends.

9.1.2. Enabling Query Analyzer by Changing MySQL Server

When enabling Query Analyzer by changing the MySQL Server, you need to shutdown your server, edit the MySQL configuration file, and then restart MySQL. You will also need to change your Agent/proxy configuration so that the Agent/proxy is listening on the original MySQL TCP/IP port. To use this method:

1. Edit the `/etc/my.cnf` or other MySQL configuration file and change or add the `port` setting from its current value (default 3306), to another value. For example:

```
port = 3307
```

2. Shutdown your MySQL Server.
3. Startup your MySQL Server and confirm that is running.
4. Edit your MySQL Enterprise Monitor Agent configuration so that the agent/proxy is listening for connections on the original MySQL port:

```
proxy-address=:3306
proxy-backend-addresses = 127.0.0.1:3307
```

5. Stop and restart MySQL Enterprise Monitor Agent.

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

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

9.2. Getting Detailed Query Information

If you click on an individual query, a pop-up window will provide more detailed information about the individual query. You can see an example of this in the figure below. The available tabs within this window will depend on whether you have configured the more detailed query information. By default, you will always be provided the Summary Details page. If enabled, you may also view Example Details, which provide more detailed data about a specific query, including the data and parameters submitted. In addition, you may also enable Example Explain, which provides you with the ability to remotely execute an `EXPLAIN` statement with the specified query and view the resulting information.

- The **Canonical Query** tab:

Figure 9.4. MySQL Enterprise Dashboard: Canonical Query Tab for a Query

Canonical Query
Example Query
Explain Query
908 B
165 B
1

Overview of information collected and aggregated for queries of this form.

Alias
None specified.

Canonical Form
[truncated](#) | [full](#) | [formatted](#)

```

SELECT
  COUNT( message_id ) AS cnt,
  DATE_FORMAT(FROM_UNIXTIME(datetime), ?) AS
  fmtdate
, process_mode, process_type
FROM
  logs_amavis
WHERE
  datetime >= ? AND datetime <= ?
GROUP BY
  process_mode ASC, process_type ASC, fmtdate
ASC

```

Execution Time Statistics

Max Time	Min Time	Avg Time	Total Time	Standard Deviation
11.123	0.274	2.140	12.838	

Row Statistics

Max Rows	Min Rows	Avg Rows	Total Rows	Standard Deviation	Total Size	Max Size
408	37	129	776	82	24.21 KB	12.8 KB

Number of Executions
6

Time Span
From Oct 27, 2008 10:16:07 AM to Oct 27, 2008 10:46:07 AM.

[expand »](#)

In addition to the summary information given in the table, you will get detailed execution statistics, including the minimum time, maximum time, average time, total time and the standard deviation. The standard deviation will enable you to determine whether a particular invocation of a query is outside the normal distribution of times for the given query.

Row statistics provide more detailed contents on the maximum, minimum, average, total, and standard deviation for the number of rows returned by the query, and the total size and maximum size of the data returned. The time period for the total and average figures is shown under the Summary Time Span.

The detailed view for a query also provides three different views of the query. The [truncated](#) version is a shortened version of the query. The [full](#) version of the query is the entire query statement. Normalization removes the constants from the indi-

vidual queries so that queries following the same logical structure are identified as the same basic query.

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

To simplify the identification of a given query, you can create a query alias. The alias will be used in place of the normalized query text within the Query Analyzer table. To create an alias for a query, click the CREATE ALIAS link against the query. The maximum length for a query alias is 255 characters.

- The **Example Query** tab:

Figure 9.5. MySQL Enterprise Dashboard: Example Query Tab for a Query

Canonical Query | **Example Query** | Explain Query | 908 B | 165 B | 134.

The query with the longest execution time during the Time Span (usually the slowest but not always).

Sampled Query
[truncated](#) | [full](#) | [formatted](#)

```
SELECT
  COUNT( message_id ) AS cnt,
  date_format( from_unixtime( datetime ), "%Y-%m-%d" )
  AS fmtdate
, process_mode, process_type
FROM
  logs_amavis
WHERE
  datetime >= 1224498843 and datetime <= 1225080000
GROUP BY
  process_mode ASC, process_type ASC, fmtdate ASC
```

Execution Time
11,122 ms

Date
Oct 27, 2008 10:35:01 AM

User
root

Thread ID
298107

From Host
192.168.0.2:22717

To Host
127.0.0.1:3306

Comments

[hide](#)

[expand »](#)

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

In addition to the full query, with data, that was executed, the tab shows the execution time, data, user, thread ID, client host and execution host for the given query.

- The **Explain Query** tab:

Figure 9.6. MySQL Enterprise Dashboard: Explain Query Tab for a Query

id	select_type	table	type	possible_keys	key	key_len	ref	rows	extra
1	SIMPLE	logs_amavis	index	null	mdatetimepmpmt	47	null	276087	Using where; Using index; Using temporary; Using filesort

The Example Explain tab enables you to view the output from running the query with the [EXPLAIN](#) prefix. For more information, see [EXPLAIN Syntax](#).

9.3. Filtering Query Analyzer Data

You can filter the queries shown within the Query Analyzer table by using the form at the top of the table. The different fields of the form are used to specify the parameters for the filter process. Once you have specified a filter, all the queries and related statistics shown within the Query Analyzer table are displayed in relation to the filter settings. For example, by default, the filter settings show the queries for the last 30 minutes. All the statistics shown are relative to the last 30 minutes, including average, maximum and execution counts.

The filter fields are:

- **SEARCH TYPE** and **QUERY SEARCH** support text searching of the normalized query. For the search type you can specify either a basic text match, or a regular expression match. In addition to the basic text match, you can also search for a query that does not contain a particular string. For regular expression searches, you can specify whether the regular expression should match, or not match (negative regexp) the queries. Regular expressions are parsed using the standard MySQL [REGEXP \(\)](#) function. For more information, see [Regular Expressions](#).

Note

The search is performed against the canonical version of the query. You cannot search against specific text or values within the parameters of the query itself.

- **DATABASE** — limit the queries to those executed within a specific database. The database match is performed using the [LIKE](#) match from the MySQL database, hence you can use the `%` and `_` characters to multiple and single character matches. For more information, see [Pattern Matching](#).
- The **TIME DISPLAY** menu selects whether the time selection for filtering should be based on the time **INTERVAL** (only queries recorded within the displayed time period are shown, using the **HOURS** and **MINUTES** popup), or whether the selection should be based on a time period (**FROM/TO**), where you can select the time range to be displayed.

Using the **INTERVAL** mode shows queries within the given time period from the point the graph was updated. For example, if you select 30 minutes, then the queries shown were captured within the last 30 minutes. If you updated the display at 14:00, then the queries displayed would have been captured between 13:30 and 14:00. Using interval mode limits the timespan for the filter selection to a maximum of 23 hours and 59 minutes.

Using the **FROM/TO** mode enables you to show queries between specific dates and times. Using this mode you can show only the queries received during a specific time span, and you can display the query history for a much longer time period, for as long as you have been recording query analysis information.

- The **VIEW** selection determines whether the information should be returned on a group basis, where an aggregate of the same query executed on all monitored servers is shown, or on a **SERVER** basis, where queries are summarized by individual server. If the latter option has been selected, the table includes an additional column showing the server.

- **QUERY TYPE** lets you select the type of query on which to filter queries. Selecting **ALL** will show all queries. Additional query types you can select include **SELECT**, **INSERT**, **UPDATE** and other main SQL query types.
- **LIMIT** specifies the number of queries to be displayed within each page.

When you have set your filter parameters, you can update the Query Analysis display by clicking the **FILTER** button. To reset the fields to the default settings click the **RESET** button.

9.4. Using Query Analyzer Data

The information provided by Query Analyzer can be complex to understand and resolve into simple targets and resolutions for your MySQL client application. The information can be used in different ways to find problems in your queries or your servers, or both. Provided below are some tips on how to get the best out of the Query Analysis interface, and how to identify different queries and problems based on the information shown by the Query Analyzer system.

First, consider the information provided by individual columns by your queries. In particular, the following columns can highlight specific problems with your queries or database server:

- **Execution Count** — High execution counts, especially for a query that you expect to be executed very rarely, may indicate that your MySQL client application is either running a simple query to frequently, or may be running a query multiple times that could otherwise be cached. You should pay particular attention to queries where the execution count increases significantly in a short period of time compared to the normal execution rate.

How to find: Use the sort feature to sort the queries by execution count.

- **New queries** — new queries appearing in the Query Analyzer tab, especially if they appear after other queries have been in the display for a number of hours or days may indicate a number of issues:
- **Execution times** — long execution times, and a long max execution time compared to the average execution time may indicate a problem with a specific query and specific parameters.

How to find: Use the sort feature to sort the queries by execution count.

You can also use the filtering and sort options to get specific information about potential problem queries.

9.5. Troubleshooting Query Analyzer

If you are having trouble with Query Analyzer, either because the information is not being shown or the full range of queries that you expect are not appearing in the Query Analyzer page then there are a number of systems you can check.

To confirm that your system is correctly configured for Query Analysis, check the following:

- Confirm that the agent is running by checking the Agent log and the status of the server within MySQL Enterprise Service Manager
- Check the configuration of the agent. You must confirm the following:
 - The `plugins` parameter within the main configuration file, `mysql-monitor-agent.ini`, must contain the `proxy` plugin:

```
plugins=proxy,agent
```

- The `agent-item-files` parameter within the main configuration file, `mysql-monitor-agent.ini`, must specify the `share/mysql-proxy/items/quant.lua` script:

```
agent-item-files = share/mysql-proxy/items/quant.lua, »
                 share/mysql-proxy/items/items-mysql-monitor.xml
```

- The proxy configuration parameters must point to the MySQL server where you want your queries to be sent. For example, if you are running your agent on the same host as your MySQL server then you might have the following lines in your `mysql-monitor-agent.ini` file:

```
proxy-address=:4040
proxy-backend-addresses = 127.0.0.1:3306
proxy-lua-script         = share/mysql-proxy/quant.lua
```

The above configuration can mean:

- The agent/proxy will listen on the current machine, using port 4040 ([proxy-address](#)).
- The agent/proxy will send all queries received on to the host 127.0.0.1 on port 3306 (the standard MySQL port), as per the [proxy-backend-addresses](#) parameter.

You can see a sample complete configuration file ([mysql-monitor-agent.ini](#)), using the 127.0.0.1 as the MySQL backend server, and reporting to a MySQL Enterprise Service Manager called [monitor](#):

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

agent-uuid = a3113263-4993-4890-8235-cadef9617c4b
log-file = mysql-monitor-agent.log
pid-file=/opt/mysql/enterprise/agent/mysql-monitor-agent.pid
```

- Confirm that you can connect through the agent proxy to your backend MySQL server. You can do this by checking with the MySQL client. You must specify the same options as you would if you were connecting to the original server, including specifying the same user and password information:

```
shell> mysql -h 127.0.0.1 --port 4040 --user=root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 299239
Server version: 5.0.60-log Gentoo Linux mysql-5.0.60-r1

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

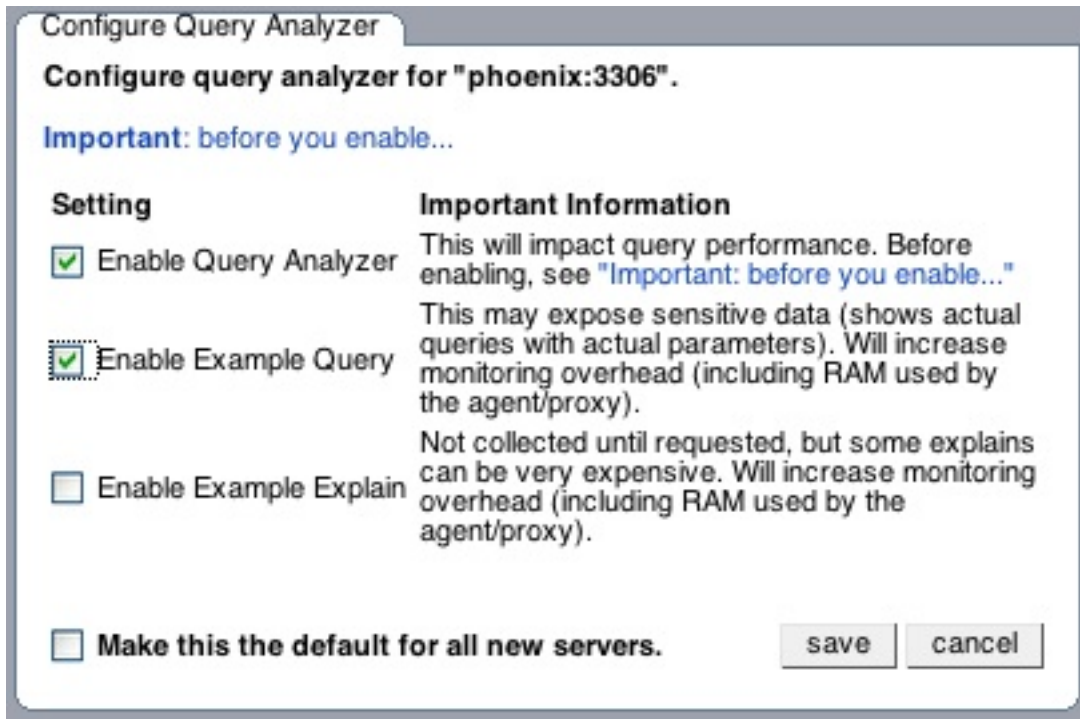
mysql>
```

- 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 9.6, “Query Analyzer Settings”](#).

9.6. Query Analyzer Settings

There are a number of settings related to the Query Analyzer data. You can configure the query analyzer operation by using the [CONFIGURE QUERY ANALYZER](#) link within the [QUERY ANALYZER](#) tab, or through the [CONFIGURE QUERY ANALYZER](#) button within the [MANAGE SERVERS](#) tab within the [SETTINGS](#) tab. Both methods provide you with the same dialog:

Figure 9.7. MySQL Enterprise Dashboard: Query Analyzer Configuration



Through either solution, the configuration options that you select are applied to the individual server or server group selected within the **SERVICES** navigation panel.

There are three configuration options available through either method:

- **ENABLE QUERY ANALYZER** configures whether query analyzer should be enabled for this server or server group. If selected, query analyzer will be enabled. To disable, delect the check box.

If Query Analyzer has been enabled, then you can additional configure the Example Query function by selecting the **ENABLE EXAMPLE QUERY** checkbox. Enabling this option provides an additional tab when you open the **CANONICAL QUERY** window when clicking on 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 9.2, "Getting Detailed Query Information"](#).

If you have enabled **EXAMPLE QUERY**, then you can additional enable **EXAMPLE EXPLAIN**. To enable this tab, select the **ENABLE EXAMPLE EXPLAIN** checkbox.

- **ENABLE EXAMPLE EXPLAIN** provides another tab when viewing a query where you can view the output from **EXPLAIN** output from MySQL for the selected query. This will show the full query and how the query was executed within the servers.

Enabling this option may add additional 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 A, MySQL Enterprise Monitor Frequently Asked Questions](#).

To enable or disable query analyzer for an individual server, go to the **SETTINGS** page and click on 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 on the current status to toggle between the On/Off position.

To disable or enable Query Analyzer for the selected servers, use the **DISABLE QUERY ANALYZER** or [enable query analyzer](#) buttons within the **SETTINGS** page. You must have selected one or more servers from the list of available servers before selecting these buttons.

You can use the options that you have just selected as the default for all new servers that register with MySQL Enterprise Service

Manager by using select the **MAKE THIS THE DEFAULT FOR ALL NEW SERVERS** checkbox. By default, when a new server registers with MySQL Monitor, the server is automatically configured to supply Query Analysis data. This can have impact on the performance of your monitor and agent as it increases the amount of information supplied to the MySQL Monitor.

Configuration of Query Analyzer occurs through the **CONFIGURE DEFAULTS** button from within the **QUERY ANALYZER** page.

Chapter 10. The Replication Page

Note

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

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

Note

Servers, whether masters or slaves, must be monitored 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 autodiscovered and a grouping is created. This grouping shows up on the replication page and also in the [Heat Chart](#) on the [Monitor](#) page. Scans run on a five minute interval, so depending upon the order of discovery, it can take as long as 2 polling intervals to create a complete group.

Figure 10.1. MySQL Enterprise Dashboard: Replication groups

Servers	Type	Slave IO	Slave SQL	Time Behind	Binlog
Replication 1 (5)	TREE	Running	Stopped		
local-tree-master:10001	master				master-bin.000035
local-tree-master-slave:10002	master/slave	Running	Running	00:00:00	master-slave-bin.000035
local-tree-slave3:10008	slave	Running	Running	00:00:00	
local-tree-slave2:10007	slave	Running	Running	00:00:00	
local-tree-slave:10003	slave	Running	Stopped		
Replication 4 (4)	MIXED	Running	Running		
ring1:10004	master/slave	Running	Running	00:00:00	master-bin.000030
ring2:10005	master/slave	Running	Running	00:00:00	master-bin.000030
ring2-slave:10009	slave	Running	Running	00:00:00	
ring3:10006	master/slave	Running	Running	00:00:00	master-bin.000030

Discovery events are logged to the [Replication](#) log. To view this log navigate to the [Settings](#) page and choose the **LOGS** link. View all replication-related events by clicking the **REPLICATION** link. This log can be a useful tool should you need to debug the replication topology discovery process.

Warning

The agent must be installed on the same machine as the server you are monitoring 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 5.3.2, "Grouping Servers"](#).

10.1. Replication Page Details

Choose a value from the **REFRESH** drop-down list box to set the rate at which information is updated. This refresh rate applies only

to the information presented on this page: It is independent of the rate set for the [Monitor](#) page.

The following columns describe replication servers and their slaves:

- Servers – Displays the group name and any master servers and slaves
- Type – Indicates the topology of a server group or in the case of individual servers, whether a server is a master, a master/slave, or a slave
- Slave IO – Reports the status of the slave 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 binlog file name
- Binlog Pos – The current position in the binlog file
- Master Binlog – The master binlog file name
- Master Binlog Pos – The current position in the master binlog 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 on 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 on a master server opens a dialog box that displays information about the server. The information shown includes:

- The number of slave servers
- The binlog file name
- The binlog 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 on a slave server opens a dialog window showing extensive information about the slave.

Appendix A. 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

- **11.1:** While monitoring my network traffic I have noticed that the agents communicate information at irregular intervals to the MySQL Enterprise Service Manager. I cannot see anything in my configuration that would explain this behaviour. What is going on?
- **11.2:** How frequently is the data purge process executed?
- **11.3:** Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?
- **11.4:** My MySQL Enterprise Service Manager is behind a firewall but it cannot communicate with the MySQL Enterprise website to register and download my license key and advisor bundle. I normally use a proxy service to access external websites. How do I configure the proxy settings for MySQL Enterprise Dashboard?
- **11.5:** What is the relationship between the advisor JAR file and the key?
- **11.6:** 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.
- **11.7:** Can the Trial-level key work with the Gold-level advisors JAR file?
- **11.8:** 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.
- **11.9:** Does the Gold-level key support Silver-level advisors?
- **11.10:** I have enabled [EXPLAIN](#) queries for Query Analyzer, but no queries with the [EXPLAIN](#) data are showing up in the display.
- **11.11:** Can I run MySQL Enterprise Service Manager on machine with other applications running?
- **11.12:** Why do some rules appear to have a **SEVERITY** of **UNKNOWN**?
- **11.13:** How do I change the name of a server?
- **11.14:** 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?

Questions and Answers

11.1: While monitoring my network traffic I have noticed that the agents communicate information at irregular intervals to the MySQL Enterprise Service Manager. I cannot see anything in my configuration that would explain this behaviour. What is going on?

Each MySQL Enterprise Monitor Agent periodically sends information to the server about a range of different information, including the core rule and statistical data, Query Analyzer information and other data used to monitor the status of your MySQL server.

One element of this is called the MySQL Enterprise Monitor Agent Heartbeat, which is core information exchange that indicates that the monitored server is still up and running. The heartbeat information is vital because it tells the MySQL Enterprise Service Manager that the agent and server are still communicating. This information is sent regularly to the MySQL Enterprise Service Manager, but to prevent multiple agents from sending the information at the same time, and creating a large network load (or storm), the interval is randomized with each heartbeat. The randomization ensures that the information is still uploaded periodically, but without the potential to overload the network with this data.

11.2: 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.

11.3: Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?

The MySQL Proxy component, and Query Analyzer, require that clients connecting through MySQL Enterprise Monitor Agent are using MySQL 5.0 or later. Clients that use the library provided with MySQL 4.1 or earlier will not work with MySQL Enterprise Monitor Agent.

11.4: My MySQL Enterprise Service Manager is behind a firewall but it cannot communicate with the MySQL Enterprise website to register and download my license key and advisor bundle. I normally use a proxy service to access external websites. How do I configure the proxy settings for MySQL Enterprise Dashboard?

To configure a proxy service, you need to edit the `apache-tomcat/conf/catalina.properties` file within the MySQL Enterprise Service Manager installation directory. To make the changes, the proxy configuration information to the end of the file by setting the `http.proxyHost` and `http.proxyPort` properties:

```
http.proxyHost=proxy.example.com
http.proxyPort=8080
```

You will need to restart the MySQL Enterprise Service Manager for the change to take effect:

```
shell> mysqlmonitorctl.sh restart
```

11.5: 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.

11.6: I have set the graphs to update every 5 minutes, and the page refresh to occur every minute. The page is refreshing correctly, but the graphs do not seem to update.

The graph refresh and page refresh are two different parameters. The graphs will update according to their refresh period, regardless of the refresh period set for the main display page.

11.7: 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.

11.8: 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 9.6, “Query Analyzer Settings”](#).

11.9: 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.

11.10: I have enabled `EXPLAIN` queries for Query Analyzer, but no queries with the `EXPLAIN` data are showing up in the display.

Query Analyzer only obtains `EXPLAIN` information when the MySQL Enterprise Monitor Agent identifies a long running query. If none of your queries exceed the defined threshold, then the `EXPLAIN` information is not obtain and provided to the Query Analyzer for display.

To change the query duration at which an `EXPLAIN` is triggered, you must edit the `share/mysql-proxy/quant.lua` file

within the MySQL Enterprise Monitor Agent directory on each server. You need to change the value configured in the `auto_explain_min_exec_time_us`. The default is 500ms:

```
---
-- configuration
--
-- SET GLOBAL analyze_query.auto_filter = 0
if not proxy.global.config.quan then
  proxy.global.config.quan = {
    analyze_queries = true,    -- track all queries
    query_cutoff   = 160,    -- only show the first 160 chars of the query
    num_worst_queries = 5,
    auto_explain   = true,
    auto_explain_min_exec_time_us = 500 * 1000
  }
end
```

The value is expressed in microseconds, which is why the value must be multiplied by 1000. To reduce this value to 100ms you would modify the line:

```
auto_explain_min_exec_time_us = 100 * 1000
```

You do not need to restart MySQL Enterprise Monitor Agent for the changes to take effect.

11.11: Can I run MySQL Enterprise Service Manager on machine with other applications running?

You can, but ideally you should be running your MySQL Enterprise Service Manager on a dedicated machine, especially if you are monitoring a number of different agents. For more information, see [Section 3.4, “Choosing Suitable MySQL Enterprise Service Manager Hardware Configurations”](#).

11.12: Why do some rules appear to have a SEVERITY of UNKNOWN?

Due to timing issues, certain rules such as “32-Bit Binary Running on 64-Bit AMD Or Intel System” and “Key Buffer Size Greater Than 4 GB” do not evaluate correctly due to timing issues. This is a known issue and will be resolved in future versions of MySQL Enterprise Monitor.

11.13: How do I change the name of a server?

Go to the **MANAGE SERVERS** panel within **SETTINGS** and click **RENAME SERVER**.

11.14: 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.

MySQL Enterprise

Questions

- [11.1: How should I decide between MySQL Enterprise Basic, Silver, Gold and Platinum?](#)
- [11.2: What is MySQL Enterprise Server?](#)
- [11.3: How do I get a 30-day trial on MySQL Enterprise?](#)
- [11.4: Does MySQL Enterprise include Maintenance, Updates, and Upgrades?](#)
- [11.5: What is a Technical Account Manager?](#)
- [11.6: Are there any Webinars available?](#)
- [11.7: What is the pricing of MySQL Enterprise?](#)
- [11.8: Does MySQL Enterprise include Emergency Hot Fix Builds?](#)
- [11.9: Are there any MySQL Enterprise White Papers available?](#)
- [11.10: How do I buy MySQL Enterprise?](#)
- [11.11: What is MySQL Enterprise?](#)
- [11.12: Does MySQL provide IP \(Intellectual Property\) Indemnification?](#)

- [11.13](#): Can I buy MySQL Enterprise subscriptions for multiple years?
- [11.14](#): What is MySQL Enterprise Unlimited?
- [11.15](#): What is MySQL Production Support?
- [11.16](#): Can I buy MySQL Enterprise subscriptions for only some of my production MySQL database servers?
- [11.17](#): Are there any Demo/Tutorials available for MySQL Enterprise?
- [11.18](#): Does MySQL Enterprise include 24x7 Technical Support?
- [11.19](#): What if I plan to add more MySQL servers to my MySQL Enterprise subscription?
- [11.20](#): What is the list of Supported Platforms?
- [11.21](#): Do all my MySQL Enterprise subscriptions need to be at the same tier?
- [11.22](#): What is MySQL Consultative Support?

Questions and Answers

11.1: 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. [Learn More](#) If you have questions and what to discuss your specific requirements, please [Contact MySQL Sales](#)

11.2: 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. [Learn More](#)

11.3: 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. [Learn More](#)

11.4: 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.

11.5: 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. [Learn More](#)

11.6: Are there any Webinars available?

Yes. MySQL provides regularly scheduled Live Webinars. [Learn More](#) 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. [Learn More](#)

11.7: 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. [Learn More](#)

11.8: 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.

11.9: Are there any MySQL Enterprise White Papers available?

Yes. Detailed architecture, technology, and business white papers are available. [Learn More](#)

11.10: 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 [Contact MySQL Sales](#)

11.11: What is MySQL Enterprise?

The MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services and production support to ensure your business achieves the highest levels of reliability, security and uptime.

MySQL Enterprise includes:

- MySQL Enterprise Server software, the most reliable, secure and up-to date version of the world's most popular open source database
- MySQL Enterprise Monitor that continuously monitors your database and proactively advises you on how to implement MySQL best practices
- MySQL 24x7 Production Support with fast response times to assist you in the development, deployment and management of MySQL applications

MySQL Enterprise is available in 4 tiers (Basic, Silver, Gold, Platinum). [Learn More](#)

11.12: 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. [Learn More](#)

11.13: 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 [Contact MySQL Sales](#)

11.14: 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. [Learn More](#)

11.15: 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. [Learn More](#)

11.16: 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 [Contact MySQL Sales](#)

11.17: Are there any Demo/Tutorials available for MySQL Enterprise?

Yes. Multiple self-running demos are available. [Learn More](#)

11.18: 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. [Learn More](#)

11.19: 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. [Learn More](#)

11.20: What is the list of Supported Platforms?

MySQL Enterprise provides broad coverage in its list of Supported Platforms. [Learn More](#)

11.21: 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.

11.22: 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. [Learn More](#)

MySQL Monitor

Questions

- [11.1: What is MySQL Enterprise Monitor?](#)
- [11.2: What versions of MySQL are supported by the MySQL Enterprise Monitor?](#)
- [11.3: What MySQL Enterprise subscription levels include the MySQL Enterprise Monitor?](#)
- [11.4: What are the features and related benefits of the MySQL Enterprise Monitor?](#)
- [11.5: What are the MySQL Enterprise Advisors and Advisor Rules?](#)
- [11.6: How is the Enterprise Monitor web application architected?](#)
- [11.7: What operating system platforms are supported by the MySQL Enterprise Monitor?](#)
- [11.8: How do I get the MySQL Enterprise Monitor?](#)
- [11.9: What makes MySQL Enterprise unique?](#)
- [11.10: What are the long-term benefits of the MySQL Enterprise Monitor?](#)
- [11.11: Which set of Enterprise Advisors, Advisor Rules and features are best for my use of MySQL?](#)
- [11.12: How is the MySQL Enterprise Monitor installed and deployed?](#)
- [11.13: What are the immediate benefits of implementing the MySQL Enterprise Monitor?](#)
- [11.14: How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?](#)
- [11.15: Which Advisors and features are included under different MySQL Enterprise subscription levels?](#)

Questions and Answers

11.1: 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. [Learn More](#)

11.2: 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.

11.3: What MySQL Enterprise subscription levels include the MySQL Enterprise Monitor?

The Enterprise Monitor is available under MySQL Enterprise subscription levels Silver, Gold and Platinum. [Learn More](#)

11.4: 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 [MySQL Enterprise Monitor Features and Benefits page](#).

11.5: What are the MySQL Enterprise Advisors and Advisor Rules?

The MySQL Enterprise Advisors are a set of best practice guidelines for the optimal use of MySQL. Advisors are spread across database specific disciplines and are comprised of a set of MySQL Advisor Rules that proactively monitor all MySQL servers and report on database application problems before they occur. Each Advisor Rule provides a detailed overview of the problem it is designed to identify, advices on how to correct the problem, specifies commands to implement the recommended fix and links to additional resources for additional research into the issue at hand. [Learn More](#)

11.6: How is the Enterprise Monitor web application architected?

The Enterprise Monitor web application is comprised of 3 components:

- **Service 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 Service 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 service 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.

11.7: 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 Service Agent supports any platform supported by the MySQL Enterprise server. For the complete list of MySQL Enterprise supported operating systems and CPUs, visit the [Supported Platforms page](#).

11.8: 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 [Trial Subscription page](#)
- To buy MySQL Enterprise, visit the [Online Shop](#)

11.9: 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.

11.10: 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.

11.11: 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.

11.12: 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.

11.13: 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

11.14: 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.

11.15: 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 [Features page](#).

MySQL Query Analyzer

Questions

- [11.1](#): Can I leave the MySQL Query Analyzer enabled at all times?
- [11.2](#): What are the typical use cases of the MySQL Query Analyzer?
- [11.3](#): What is the MySQL Query Analyzer?
- [11.4](#): What are the main features and benefits of the MySQL Query Analyzer?
- [11.5](#): What makes the MySQL Query Analyzer unique?
- [11.6](#): How are subscribers notified about updates to the MySQL Query Analyzer application components?

- [11.7: How can I get the MySQL Query Analyzer?](#)
- [11.8: How is the MySQL Query Analyzer installed and enabled?](#)
- [11.9: What overhead can I expect when the MySQL Query Analyzer is installed and enabled?](#)

Questions and Answers

11.1: 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 non-peak hours, typically during 30 minute windows. They can then view the collected queries using the date/time or interval filter options.

11.2: 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 non-peak hours.

11.3: 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. [Learn More](#)

11.4: What are the main features and benefits of the MySQL Query Analyzer?

For the complete features and benefits, visit the [MySQL Enterprise Monitor Features and Benefits page](#).

11.5: 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.

11.6: 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.

11.7: 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 [Trial Subscription page](#)
- To buy MySQL Enterprise, visit the [Online Shop](#)

11.8: How is the MySQL Query Analyzer installed and enabled?

The Query Analyzer feature is installed with the Service 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.

11.9: 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.

Appendix B. Files Associated with The MySQL Enterprise Monitor

Note

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

This section details the files associated with the MySQL Enterprise Monitor and shows their default location.

B.1. Log Files for the MySQL Enterprise Service Manager

This section shows the location of the log and configuration files associated with the various components that make up the MySQL Enterprise Service Manager. These files can prove useful for debugging purposes.

A table showing the log file locations on Windows follows.

Table B.1. MySQL Enterprise Monitor: Log file locations – Windows

Component	File Location
Apache/Tomcat	C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\logs
Repository	C:\Program Files\MySQL\Enterprise\Monitor\mysql\data
Configuration Report	C:\Program Files\MySQL\Enterprise\Monitor\configuration_report.txt

A table showing the log file locations on Unix follows.

Table B.2. MySQL Enterprise Monitor: Log file locations – Unix

Component	File Location
Apache/Tomcat	/opt/mysql/enterprise/monitor/apache-tomcat/logs
Repository	/opt/mysql/enterprise/monitor/mysql/data
Configuration Report	/opt/mysql/enterprise/monitor/configuration_report.txt

Find below a table showing the log file locations on Mac OS X.

Table B.3. MySQL Enterprise Monitor: Log file locations – Mac OS X

Component	File Location
Apache/Tomcat	/Applications/mysql/enterprise/monitor/apache-tomcat/logs
Repository	/Applications/mysql/enterprise/monitor/mysql/data
Configuration Report	/Applications/mysql/enterprise/monitor/configuration_report.txt

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.

B.2. Monitor Agent and Service Manager Installation Log Files

These log files are called `bitrock_installer.log`.

On Windows these files are stored in the temporary directory defined by the variable `%TEMP%`. To determine the value of `%TEMP%` on your system, type `echo %TEMP%` at the command line. Alternately, you may open the Run command window, enter `%TEMP%`

and press OK. To find this file on Unix and Mac OS X check the value of the environment variable, `$TMPDIR`. If this variable is not defined, look in the `/tmp` directory.

If the file `bitrock_installer.log` already exists, a file called `bitrock_installer_xxxx.log` (where `xxxx` stands for an arbitrary series of numerals) will be created.

To determine if an installation file belongs to the Monitor Agent or the Service Manager you must open the file in a text editor.

B.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 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.

B.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 B.4. MySQL Enterprise Monitor: MIB file locations

Operating System	File Location
Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\support-files\MONITOR.MIB</code>
Unix	<code>/opt/mysql/enterprise/monitor/support-files/MONITOR.MIB</code>
Mac OS X	<code>/Applications/mysql/enterprise/monitor/support-files/MONITOR.MIB</code>

B.5. The `config.properties` File

The repository user name and password are stored in the `config.properties` file. A table showing the location of this file follows:

Table B.5. MySQL Enterprise Monitor: The `config.properties` file

Operating System	File Location
Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\webapps\WEB-INF</code>
Unix	<code>/opt/mysql/enterprise/monitor/apache-tomcat/webapps/WEB-INF</code>
Mac OS X	<code>/Applications/mysql/enterprise/monitor/apache-tomcat/webapps/WEB-INF</code>

You should make sure that the file is secured at the file system level so that it cannot be read by anybody but the administrator or

Enterprise monitor.

Appendix C. Error codes

The following table documents error codes reported by the GUI and the associated message.

Table C.1. MySQL Enterprise Dashboard: GUI Error Codes and Messages

Error Code	Message
U0120	Unable to import Advisor Jar. You may download the jar manually from the Enterprise website and import it on this page.
U0121	Unable to import subscription key. You may download the key manually from the Enterprise website and import it on this page.

The following table documents error codes reported by the Enterprise server and the associated message.

Table C.2. MySQL Enterprise Monitor: Server Codes and Messages

Error Code	Message
E0001	Internal Error: <i>replaceable_value</i>
E0100	Invalid user role " <i>replaceable_value</i> ". Only "manager", "dba", and "agent" are valid.
E0101	Invalid user name " <i>replaceable_value</i> ". 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 " <i>replaceable_value</i> " already exists.
E0106	User " <i>replaceable_value</i> " does not exist.
E0107	User role " <i>replaceable_value</i> " is not authorized.
E0200	No mapping exists to map the user " <i>replaceable_value</i> " to a MySQL Network user.
E0201	A mapping already exists for the user " <i>replaceable_value</i> " 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: " <i>replaceable_value</i> ".
E0205	Your Advisors are currently up to date.
E0206	Your MySQL Enterprise trial expired after " <i>replaceable_value</i> ".
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: <i>replaceable_value</i> Requires role: <i>replaceable_value</i>
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: <i>replaceable_value</i> .
E0300	Can not delete default group.
E0301	Group " <i>replaceable_value</i> " already exists.
E0302	Group " <i>replaceable_value</i> " does not exist.
E0303	Invalid group name " <i>replaceable_value</i> ". Group name must be alphanumeric and less than 128 characters.
E0304	Group element " <i>replaceable_value</i> " already exists.
E0305	Group element " <i>replaceable_value</i> " does not exist in the group " <i>replaceable_value</i> ".
E0400	Agent " <i>replaceable_value</i> " does not exist.
E0401	Agent payload parameter NULL.
E0402	Agent task " <i>replaceable_value</i> " does not exist.

Error Code	Message
E0403	There has been a problem de-serializing this agent's request. Please ensure all agents are version <i>replaceable_value</i> or greater.
E0500	Server " <i>replaceable_value</i> " already exists.
E0501	Server " <i>replaceable_value</i> " does not exist.
E0502	Server " <i>replaceable_value</i> " is actively being monitored and cannot be deleted.
E0503	Server " <i>replaceable_value</i> " is being migrated and cannot be deleted. Please halt the active migration first.
E0600	Database " <i>replaceable_value</i> " already exists.
E0601	Database " <i>replaceable_value</i> " does not exist.
E0700	Threshold " <i>replaceable_value</i> " already exists.
E0701	Threshold " <i>replaceable_value</i> " 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 occurring for " <i>replaceable_value</i> ".
E0802	Data collection does not exist for " <i>replaceable_value</i> ".
E0803	Invalid lifespan. Data life must be greater than 0 days.
E0804	Invalid namespace. Supported namespaces include "mysql" and "os". <i>replaceable_value</i>
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: " <i>replaceable_value</i> ". 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 " <i>replaceable_value</i> " 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. " <i>replaceable_value</i> " Please see the error logs for details.
E0813	Invalid server id " <i>replaceable_value</i> ".
E0814	No data items exist for server " <i>replaceable_value</i> ".
E0815	Value " <i>replaceable_value</i> " not valid for " <i>replaceable_value</i> ".
E0816	categoryName has to be set
E0817	You must specify either a server or group in the data collection schedule.
E0818	The server " <i>replaceable_value</i> " does not support item(s) " <i>replaceable_value</i> " required by monitor " <i>replaceable_value</i> ".
E0819	Invalid data type: " <i>replaceable_value</i> ".
E0900	Advisor id " <i>replaceable_value</i> " does not exist.
E0901	Advisor name " <i>replaceable_value</i> " does not exist.
E0902	Advisor XML is invalid.
E0903	Category " <i>replaceable_value</i> " does not exist.
E0904	Invalid server list.
E0905	SKIPPED " <i>replaceable_value</i> " for <i>replaceable_value</i> because Agent for <i>replaceable_value</i> is down.
E0906	SKIPPED " <i>replaceable_value</i> " for <i>replaceable_value</i> because mysqld on <i>replaceable_value</i> is down.
E0907	Category " <i>replaceable_value</i> " cannot be modified.
E0908	Category " <i>replaceable_value</i> " cannot be deleted.
E0909	Category " <i>replaceable_value</i> " already exists.
E0910	Advisor bundle version <i>replaceable_value</i> is incompatible with server version <i>replaceable_value</i> .

Error Code	Message
E0911	Heat chart rule " <i>replaceable_value</i> " scheduled against " <i>replaceable_value</i> " may not be un-scheduled.
E0912	Heat chart rule " <i>replaceable_value</i> " scheduled against " <i>replaceable_value</i> " may not be disabled.
E1100	Graph time format " <i>replaceable_value</i> " is invalid. Must be in the format HH:MM:SS.
E1101	Graph size height and width must be specified.
E1102	Graph " <i>replaceable_value</i> " does not exist.
E1103	Graph " <i>replaceable_value</i> " already exists.
E1104	Graph " <i>replaceable_value</i> " contains a data collection item that can not be identified.
E1200	Failed to retrieve identity column.
E1201	Required parameter " <i>replaceable_value</i> " was NULL.
E1202	Only SELECT and SHOW commands are allowed via this interface.
E1203	Invalid timezone " <i>replaceable_value</i> ".
E1204	Invalid interval " <i>replaceable_value</i> ".
E1205	Could not find object " <i>replaceable_value</i> " in cache " <i>replaceable_value</i> ".
E1206	Feature is not implemented. Parameters: <i>replaceable_value</i>
E1300	Advisor " <i>replaceable_value</i> " already exists.
E1301	Advisor " <i>replaceable_value</i> " does not exist.
E1302	Schedule does not exist. " <i>replaceable_value</i> " " <i>replaceable_value</i> "
E1303	JEP error: expression : " <i>replaceable_value</i> ", message: " <i>replaceable_value</i> ".
E1304	Advisor " <i>replaceable_value</i> " contains a data collection item that can not be identified.
E1305	Alarm level " <i>replaceable_value</i> " is not valid.
E1306	Schedule already exists. " <i>replaceable_value</i> " " <i>replaceable_value</i> ".
E1307	Advisor " <i>replaceable_value</i> " is a read only MySQL Network certified Advisor. Please copy the rule before editing.
E1308	Advisor " <i>replaceable_value</i> " is currently scheduled against one or more monitored MySQL servers and cannot be removed.
E1309	Could not render text. " <i>replaceable_value</i> ".
E1310	No open event for " <i>replaceable_value</i> " on server " <i>replaceable_value</i> ". Perhaps already closed.
E1311	Alarm with id " <i>replaceable_value</i> " does not exist.
E1400	Invalid missed agent heartbeat threshold value.
E1401	Notification entry " <i>replaceable_value</i> " does not exist.
E1402	Agent " <i>replaceable_value</i> " is using session " <i>replaceable_value</i> " but session " <i>replaceable_value</i> " was requested.
E1403	Server name " <i>replaceable_value</i> " is in use by another agent with uuid " <i>replaceable_value</i> ".
E1500	Notification target " <i>replaceable_value</i> " does not exist.
E1501	Invalid email address " <i>replaceable_value</i> ".
E1502	Notification target " <i>replaceable_value</i> " already exists.
E1503	Invalid notification target name " <i>replaceable_value</i> ".
E1504	Invalid monitor name, it must not be empty.
E1505	No variable assignments given, you must define at least one.
E1506	SMTP authentication failed.
E1507	SMTP send failed
E1508	Invalid SNMP target " <i>replaceable_value</i> ".
E1509	Invalid SNMP trap type " <i>replaceable_value</i> ".
E1600	Log " <i>replaceable_value</i> " does not exist.
E1700	Authentication Mechanism was null
E1701	Unsupported Authentication Mechanism <i>replaceable_value</i>
E1702	Incorrect password for user <i>replaceable_value</i>

Error Code	Message
E1800	Invalid data type: " <i>replaceable_value</i> ", new value: " <i>replaceable_value</i> ".
E1801	Invalid value: " <i>replaceable_value</i> " for data type " <i>replaceable_value</i> " for item " <i>replaceable_value</i> ".
E1802	Invalid expression: " <i>replaceable_value</i> ", exception: " <i>replaceable_value</i> ", raw expression: " <i>replaceable_value</i> ". substitute values: " <i>replaceable_value</i> ".
E1900	History not found id: " <i>replaceable_value</i> ", schedule: " <i>replaceable_value</i> ", expression: " <i>replaceable_value</i> ".
E2000	Tag not found, id: " <i>replaceable_value</i> ".
E2101	Unable to collect " <i>replaceable_value</i> " on " <i>replaceable_value</i> " for instance " <i>replaceable_value</i> ".
E9000	<i>replaceable_value</i>
E9001	<i>replaceable_value</i> from: <i>replaceable_value</i>

Appendix D. 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.

D.1. Changes in MySQL Enterprise Monitor 2.1.0 (Not yet released)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.6.

Functionality added or changed:

- The agent should be able to run as a non-root user. However, the startup scripts always started it as root.

The agent chassis now has a new option `--user` to drop privileges after being started as root. Note, this does not work when not started as superuser, nor on Windows.

A new dialog box has also been added to the agent installer. The dialog has the following text: “The agent does not need to run with root user privileges. The agent will switch to the user account provided below when started by the root user.”

The dialog also has a text field to allow the entry of the user account.

A new parameter was also added to the `mysql-monitor-agent.ini` file. The parameter has the format `user=xxx`, where xxx is the user account to be used. ([Bug#33778](#))

Bugs fixed:

- The recommendation for the rule `Table Cache Not Optimal` says:

```
Recommended Action
SET GLOBAL table_cache = (64 + 16);
```

But an error was generated on executing that query:

```
mysql> SET GLOBAL table_cache = (64 + 16);
ERROR 1193 (HY000): Unknown system variable 'table_cache'

mysql> select version();
+-----+
| version() |
+-----+
| 5.1.31-log |
+-----+
1 row in set (0.01 sec)
```

([Bug#44602](#))

- The Mac OS X version of the Service Manager uses the system JRE. The system JRE loads the libraries located in `/Library/Java/Extensions`. As libraries in the extensions directory take precedence over other libraries, this caused conflicts when user extension libraries were installed there, as these would be used by the JRE when running Service Manager, instead of the shipped libraries. This happened when Java-related products were installed such as Connector/J, Spring, and Hibernate.

This fix stops user-installed extension libraries from being used when the JRE runs the Service Manager, thus giving a “pristine” environment with no library collisions. ([Bug#44157](#))

- Accessing the Query Analyzer tab caused a full table scan to take place on the MySQL Enterprise Monitor database. ([Bug#43989](#))
- The Replication Group was renamed back to its default name after a new topology was discovered. ([Bug#43816](#))
- A new topology was not discovered after the previous replication group was renamed. ([Bug#43815](#))
- On Unix systems, executing the command:

```
./mysqlmonitorctl.sh stop
```

did not make sure that `mysqld` was shutdown before finishing.

This resulted in a situation such as the following:

```
# /opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh stop
Using CATALINA_BASE: /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_HOME: /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_TMPDIR: /opt/mysql/enterprise/monitor/apache-tomcat/temp
Using JRE_HOME: /opt/mysql/enterprise/monitor-2.0.0.7092/java
Stopping tomcat service ... [ OK ]
/opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh : mysql stopped
```

However, running the following command a few minutes later showed that the MySQL server was still running:

```
# /opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh status
MySQL Network MySQL is running
MySQL Network Tomcat is not running
```

([Bug#43803](#))

- The MySQL Enterprise Monitor upgrade installer incorrectly replaced the `AdvisorScript.jar` in `<instDir>/apache-tomcat/webapps/ROOT/WEB-INF/lib/` with the default Advisor JAR. ([Bug#43773](#))
- The SNMP trap source IP was always set to 127.0.0.1. ([Bug#43738](#))
- The installer exited with a return code 0, even if an error was detected and reported to the user during the installation.

```
../mysqlmonitor-2.1.0.1015-linux-x86-64bit-installer.bin --mode unattended --installdir
/data0/merlin/monitoring/2.1.0.1015/host/38080 --tomcatport 38080 --tomcatshutdownport
38503 --tomcatsslport 38443 --dbport 33300 --usessl 1 --adminuser **user** --adminpassword **pwd**
Error running /data0/merlin/monitoring/2.1.0.1015/host/38080/mysql/bin/mysql
--defaults-file=/data0/merlin/monitoring/2.1.0.1015/host/38080/mysql/my.cnf -S
/data0/merlin/monitoring/2.1.0.1015/qa-merlin/38080/mysql/tmp/mysql.sock -u root -D mysql
-e "update user set Password = PASSWORD('**pwd**') where User = 'root'; update user set
User = '**user**' where User = 'root';delete from user where User = '';flush privileges;"
: ERROR 2002 (HY000): Can't connect to local MySQL server through socket
'/data0/merlin/monitoring/2.1.0.1015/host/38080/mysql/tmp/mysql.sock' (2)
```

([Bug#43676](#))

- The agent created the `mysql.inventory` table with an engine type of InnoDB, instead of MyISAM, when InnoDB was specified as the default engine type in `my.cnf`. This happened because the agent did not explicitly specify the table engine type to be of MyISAM. ([Bug#43551](#))
- When a trailing space, or tab character, was added at the end of a parameter in the `config.properties` file, MySQL Enterprise Monitor failed to start and generated the following errors in log:

```
ERROR [Thread-1:org.springframework.web.context.ContextLoader] Context initialization
failed
...
Caused by: java.sql.SQLException: Illegal connection port value '13306'
...
```

Resolving the problem required detailed log analysis because the configuration file did not show any apparent problems. ([Bug#43540](#))

- If a host was not a slave during the initial discovery phase, then it would not be displayed in the Replication tab if it subsequently became a slave.

This was because after the initial discovery phase, if a host did not have `slavestatus` present, no subsequent checks were made to check for the host being a slave. It was therefore missed for the purposes of replication discovery and never showed in the Replication tab. ([Bug#42997](#))

- Heat Chart rules could not be disabled or unscheduled. ([Bug#42932](#))
- There were four columns added to the `SHOW SLAVE STATUS` query in MySQL Server 5.1: `Last_IO_Errno`, `Last_IO_Error`, `Last_SQL_Errno`, and `Last_SQL_Error`.

However, these were not displayed within the Replication tab. ([Bug#42811](#))

- The installer used to upgrade from version 1.3 corrupted passwords containing the “?” character. ([Bug#42452](#))
- Sun multi-core processors caused all cores to be reported on the meta information page.

The larger T-series SPARC processors have 32+ cores. This caused the meta information page in the Dashboard to scroll as it reported each one. (Bug#42355)

- The username field for new users was populated by the last username used.

When creating a new user for the second time in Dashboard, the previously created username appeared in the dialog. (Bug#42314)

- The `my.cnf` file for the Enterprise Monitor internal database had the following configuration item:

```
innodb_autoextend_increment = 50M
```

This generated the error:

```
16:36:23 [Warning] option 'innodb_autoextend_increment': unsigned value 52428800
adjusted to 1000
```

This variable is interpreted as being specified in MB, so 50M would be 50 TB. Such a high value results in the variable being adjusted to 1000 MB.

The value in the configuration file should be:

```
innodb_autoextend_increment = 50
```

(Bug#42096)

- A number of Advisor rules had advice text that had not been translated into Japanese. The Advisors that contained untranslated rules included Performance, Schema and Security. (Bug#42067)
- `OM_REFRESH` was not supported by MySQL Proxy, it caused an `abort()`.

```
shell> ./mysql-proxy --proxy-backend-addresses=192.168.250.3:3306
network-mysqld-proxy.c.3524: COM_(0x07) is not handled
Aborted (core dumped)

(gdb) bt
#0 0x00b1b402 in ?? ()
#1 0x00cbaf30 in raise () from /lib/i686/noseg/libc.so.6
#2 0x00cbc811 in abort () from /lib/i686/noseg/libc.so.6
#3 0x08061efc in IA__g_logv at gmessages.c:497
#4 0x08061f66 in IA__g_log at gmessages.c:517
#5 0x08054645 in proxy_read_query_result at network-mysqld-proxy.c:3522
#6 0x0804c5f4 in plugin_call at network-mysqld.c:977
#7 0x0804d45a in network_mysqld_con_handle at network-mysqld.c:1520
#8 0x08057cb9 in event_process_active (base=0x978b260) at event.c:331
#9 0x08057e64 in event_base_loop (base=0x978b260, flags=0) at event.c:449
#10 0x08057d1c in event_base_dispatch (event_base=0x978b260) at event.c:351
#11 0x0804d9d0 in network_mysqld_thread (_srv=0x9789008) at network-mysqld.c:1768
#12 0x0804b84a in main (argc=1, argv=0xbfc4fe84) at mysql-proxy.c:615
```

(Bug#41991)

- The MySQL Enterprise Monitor file `my.cnf` specified an initial size of 500M for the central tablespace. However, `innodb_file_per_table` was used as well, resulting in approximately 500M of space being potentially wasted. (Bug#41967)
- After an error was generated due to an incorrect password while trying to create a new user, the following error was obtained when subsequently attempting to create a valid new user:

```
U0002 You must log in to access the requested resource
```

(Bug#41384)

- SNMP trap messages were sending 127.0.0.1 as the IP address, and there was no feature to allow the user to configure the IP address contained in the SNMP message, which would have been useful for troubleshooting. (Bug#41361)
- Allowing the heat chart rules to be set to unscheduled caused the user interface to appear broken. (Bug#41312)
- Graphs were incorrect for data that did not change. The graphs appeared as if no data had been gathered.

The Hit Ratios graph had gaps in it where there had not been any activity on the parameters being monitored. For instance, if MyISAM tables were not used, then no Key Cache hit ratio series was plotted, even though the variables were still being collected. (Bug#41232)

- When creating a new Database Administrator user in FireFox 2 the following error message was generated:

```
U0002 You must log in to access the requested resource.
```

This occurred in a new installation using the default administrator account. No Query Analysis permissions were given. However, the operation worked correctly using the Safari web browser. ([Bug#41032](#))

- The **MANAGE SERVERS** page did not refresh in a manner consistent with other pages. This meant that changes to configuration made by others would not be reflected on the page. Also, changes in the status of the servers were not displayed automatically. ([Bug#40792](#))
- The agent installer for Solaris 8 x86 32-bit was missing. ([Bug#40248](#))
- Even though Query Analysis was disabled through the user interface, the queries that go through the agent were still being collected.

When Query Analysis was turned back on in the user interface, those queries were then displayed. ([Bug#40032](#))

- Alerts sent from MySQL Enterprise Monitor used the GMT timezone, for example:

```
Time: 2008-09-17 19:41:08 GMT
```

That was not convenient for users, as their timezones may not have been GMT. ([Bug#39504](#))

- The MySQL Enterprise Monitor upgrade installer replaced the `my.cnf` file. This resulted in the loss of any changes that had been made to the configuration file. ([Bug#36528](#))
- If the “On Save send test trap” checkbox was checked when the SAVE button was clicked and the locale was set to Japanese, an error occurred. The orange error banner was displayed at the top of the page with the error message in Japanese. ([Bug#32069](#))

D.2. Changes in MySQL Enterprise Monitor 2.0.6 (Not yet released)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.5.

Bugs fixed:

- An error was generated when an attempt was made to rename a Replication Group to one that had previously been deleted. ([Bug#43846](#))
- A new topology was not discovered after the previous replication group was renamed. ([Bug#43815](#))
- On Unix systems, executing the command:

```
./mysqlmonitorctl.sh stop
```

did not make sure that `mysqld` was shutdown before finishing.

This resulted in a situation such as the following:

```
# /opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh stop
Using CATALINA_BASE:   /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_HOME:   /opt/mysql/enterprise/monitor/apache-tomcat
Using CATALINA_TMPDIR: /opt/mysql/enterprise/monitor/apache-tomcat/temp
Using JRE_HOME:        /opt/mysql/enterprise/monitor-2.0.0.7092/java
Stopping tomcat service ... [ OK ]
/opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh : mysql stopped
```

However, running the following command a few minutes later showed that the MySQL server was still running:

```
# /opt/mysql/enterprise/monitor-2.0.0.7092/mysqlmonitorctl.sh status
MySQL Network MySQL is running
MySQL Network Tomcat is not running
```

([Bug#43803](#))

- The MySQL Enterprise Monitor upgrade installer incorrectly replaced the `AdvisorScript.jar` in `<instDir>/apache-tomcat/webapps/ROOT/WEB-INF/lib/` with the default Advisor JAR. ([Bug#43773](#))

- The agent created the `mysql.inventory` table with an engine type of InnoDB, instead of MyISAM, when InnoDB was specified as the default engine type in `my.cnf`. This happened because the agent did not explicitly specify the table engine type to be of MyISAM. (Bug#43551)
- If a host was not a slave during the initial discovery phase, then it would not be displayed in the Replication tab if it subsequently became a slave.

This was because after the initial discovery phase, if a host did not have `slavestatus` present, no subsequent checks were made to check for the host being a slave. It was therefore missed for the purposes of replication discovery and never showed in the Replication tab. (Bug#42997)

D.3. Changes in MySQL Enterprise Monitor 2.0.5 (18th March 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.4.

Bugs fixed:

- The FreeBSD 7 Agent was inadvertently a Linux binary.

```
shell> file mysqlmonitoragent-2.0.5.7153-freebsd7-x86-32bit-installer.bin
mysqlmonitoragent-2.0.5.7153-freebsd7-x86-32bit-installer.bin: ELF 32-bit LSB executable,
Intel 80386, version 1 (SYSV), for GNU/Linux 2.2.5, dynamically linked (uses shared libs),
stripped
```

(Bug#44263)

- Calling the Agent with the option `--agent-run-os-tests` resulted in a crash. This happened on Linux x86-64 systems. The resultant stack trace was:

```
(ga-merlin)
2009-03-04 16:39:42: (critical) chassis.c:1097: could not raise RLIMIT_NOFILE to 8192,
Invalid argument (22). Current limit still 1024.
sigar-test-all.c.124 (test_sigar_pid_get):
  pid = 5188
sigar-test-all.c.106 (test_sigar_mem_get):
  ...cut...
sigar-test-all.c.427 (test_sigar_file_system_list_get): (items = 13)
[0]
  fs.dirname = /
  fs.devname = /dev/mapper/vg00-root
  fs.type = local
  fs.sys-type-name = ext3
  fs.type = 2
  fsusage.total = 15481840
  fsusage.free = 14116140
  fsusage.used = 1365700
  fsusage.avail = 13329708
  fsusage.files = 1966080
  fsusage.use_percent = 0.100000
[0]
  diskusage.reads = 315302
  diskusage.writes = 6318240
  diskusage.write_bytes = 25879511040
  diskusage.read_bytes = 6561092608
  diskusage.queue = 47457530080206
Segmentation fault
```

On some systems no output was shown, other than the message “Segmentation fault”. (Bug#43381)

- Following a change in the replication configuration, MySQL Enterprise Monitor did not display the new topology correctly. (Bug#43240)
- When a data collection became invalid, the agent sent NULLs for those collection values. However, the timestamps that it sent with the values were the timestamps from the last valid value that was collected.

Due to key constraints on the Service Manager side, MySQL Enterprise Monitor disregarded anything sent with duplicate timestamps, thus the NULLs received from the agent were never processed. Some mechanisms, such as replication discovery, depend on the NULLs to identify a situation where data has become invalid. (Bug#43239)

- MySQL Enterprise Monitor did not add a log entry each time data was purged. The log entry should have noted how many rows of each type of data were purged (historical data, logs, quan data). (Bug#43159)
- MySQL Enterprise Monitor generated a stack overflow. This was as a result of a bug in Hibernate, which caused Hibernate to enter infinite recursion while trying to load a relationship. (Bug#43107)

- The “Table Cache Set Too Low For Startup” and “Table Cache Not Optimal” rules were not supported on MySQL 5.1 because the `table_cache` system variable was deprecated and replaced with `table_open_cache`. (Bug#42663)
- Migrated server was not completely deleted.

In a Monitor that had been updated from 1.3.2 to 2.0.4, with 2 database servers queued for migration, if a server being migrated was deleted, or a migrated server was deleted, this would not be reflected in the user interface or in the license count, until Tomcat was restarted. (Bug#42604)

- The installer used to upgrade from version 1.3 corrupted passwords containing the “?” character. (Bug#42452)
- Sun multi-core processors caused all cores to be reported on the meta information page.

The larger T-series SPARC processors have 32+ cores. This caused the meta information page in the Dashboard to scroll as it reported each one. (Bug#42355)

- If an attempt was made to disable a rule using the link next to the rule, the following error message was generated:

```
U0002 You must log in to access the requested resource. Go to login page.
```

However, clicking on the link did not prompt the user to login again. (Bug#42313)

- Changing `ssh-agent` from OpenSSH or specifying a malevolent value of `agent-host-id`, could inject data into the monitored MySQL Server.

For example, setting `agent-host-id` to the value “I’m a test” would result in the following message in the error log:

```
2009-01-23 15:45:11: ((error)) agent_mysql_d.c:281: mysql_real_query('INSERT INTO
mysql.inventory (name, value) VALUES ( 'hostid', 'I'm a test' )') on 'mysql' failed: You
have an error in your SQL syntax; check the manual that corresponds to your MySQL server
version for the right syntax to use near 'm a test' )' at line 1 (mysql-errno = 1064)
```

(Bug#42306)

- When `SHOW GLOBAL STATUS` returned a value greater than 214748364, it was sent to the Service Manager as 214748364. (Bug#42245)
- The Agent failed to identify local sockets as local on Mac OS X 10.4.

If the Agent was configured to use a Unix domain socket on Mac OS X 10.4, it did not treat the connection as local and failed to provide CPU and memory information to MySQL Enterprise Monitor. This is shown in the log file:

```
2009-01-20 18:15:02: (message) network-socket.c:752: is-local family 0 != 1
2009-01-20 18:15:02: (message) agent_mysql_d.c:322: [mysql] mysql_d is not local or not
directly connected
```

However, this problem did not happen on Mac OS X 10.5. (Bug#42220)

- Some graphs on the **GRAPH** tab were not updated after the page was refreshed, or UPDATE was clicked.

The only way to get an updated graph was to change the graph size (in pixels) and then click UPDATE. (Bug#42162)

- The `my.cnf` file for the Enterprise Monitor internal database had the following configuration item:

```
innodb_autoextend_increment = 50M
```

This generated the error:

```
16:36:23 [Warning] option 'innodb_autoextend_increment': unsigned value 52428800
adjusted to 1000
```

This variable is interpreted as being specified in MB, so 50M would be 50 TB. Such a high value results in the variable being adjusted to 1000 MB.

The value in the configuration file should be:

```
innodb_autoextend_increment = 50
```

(Bug#42096)

- A number of Advisor rules had advice text that had not been translated into Japanese. The Advisors that contained untranslated rules included Performance, Schema and Security. ([Bug#42067](#))
- The Service Manager did not handle the case where the agent failed to supply a valid `master_ip`. The Service Manager would then not restart. The logs contained the following:

```
2009-01-09 14:39:50,472 ERROR [main:org.springframework.web.context.ContextLoader] Context
initialization failed
org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with
name 'serverConnectionMonitor' defined in ServletContext resource
[/WEB-INF/applicationContext.xml]: Unsatisfied dependency expressed through constructor
argument with index 2 of type [com.mysql.etoools.monitor.bo.Manager]: Error creating bean
with name 'manager' defined in ServletContext resource [/WEB-INF/applicationContext.xml]:
Invocation of init method failed; nested exception is
com.mysql.etoools.monitor.pom.UnsupportedAttributeException:
101c6b5b-15eb-49aa-916c-843c51b28d38: mysql.slavestatus.Master_ip
```

([Bug#42005](#))

- Having too many users with strong privileges generated Service Manager errors and events failed to be triggered.

If there were approximately 1000 users with full privileges and the value of `group_concat_max_len` was set to 100001, the size of the data that the agent sent to the Service Manager was too large and caused Service Manager errors. Also, the Security event “Account has Strong MySQL privileges” did not trigger. ([Bug#41987](#))

- Query Analyzer's Query Type filter for `SELECT` ignored statements starting with parenthesis.

If you sent a statement through Query Analyzer starting with parenthesis, such as:

```
(SELECT 1 FROM dual);
```

and then attempted to filter for `SELECT` queries only, queries starting with parenthesis were not displayed. ([Bug#41968](#))

- The agent angel process was spawning too soon, which caused a duplicate UUID error.

`g_error()` logged a fatal error, and called `abort()`. This terminated the program. Then the angel respawned with the same UUID, but with a -1 session resync request, which the MEM server denied because the old session was still active. This resulted in a permissions denied error (1142) on the `mysql.inventory` table.

```
2008-12-17 18:58:58: (message) agent_mysqlqd.c:313: [mysql] mysqlqd is local and directly connected
2008-12-17 18:58:58: ((error)) agent_mysqlqd.c:444: [mysql] mysql_real_query("SELECT value
FROM mysql.inventory WHERE name = 'uuid'") failed: SELECT command denied to user
'ent_agent'@'127.0.0.1' for table 'inventory' (errno=1142)
2008-12-17 18:58:58: (debug) chassis.c:282: 15134 returned: 15134
2008-12-17 18:58:58: (message) chassis.c:304: [angel] PID=15134 died on signal=6 (it used
0 kBytes max) ... waiting 3min before restart
2008-12-17 18:59:00: (debug) chassis.c:244: we are the child: 15149
2008-12-17 18:59:00: (message) chassis.c:259: [angel] we try to keep PID=15149 alive
2008-12-17 18:59:00: (debug) chassis.c:277: waiting for 15149
2008-12-17 18:59:00: (message) mysql-proxy 0.7.0 started
2008-12-17 18:59:00: (message) MySQL Monitor Agent 2.0.0.7111 started.
```

([Bug#41600](#))

- `master_uuid` discovery did not work with MySQL Server versions prior to 5.1. `master_uuid` did not show in any Agent to Monitor communications, and no log or error messages were generated.

However, now the bug has been fixed, an Agent monitoring a 5.0 MySQL Server would register the following in its logs:

```
...
<classname>slavestatus</classname>
<instance>12515cdc-8c00-4223-9d2a-2666a403512c</instance>
<attribute>Master_uuid</attribute>
</target>
<utc>2009-03-03T19:58:05.700Z</utc>
<value>b2fd9f86-6e42-49f2-b930-e8fb3e728179</value>
```

Note the presence of `master_uuid`. ([Bug#41525](#))

- The `master_uuid` was not used for replication topology discovery.

The agent collected `master_uuid` by reading the `master.info` file, and then running a `SELECT` directly against its master. This was to try and read the `mysql.inventory` table on the master to obtain the instance `master_uuid`.

However, this was not being used correctly by the replication topology discovery within the server. ([Bug#41519](#))

- Queries such as `SELECT` against the master `mysql.inventory` was not logged on slave-based agents. This made diagnosing topology discovery issues difficult. (Bug#41518)
- After an error was generated due to an incorrect password while trying to create a new user, the following error was obtained when subsequently attempting to create a valid new user:

```
U0002 You must log in to access the requested resource
```

(Bug#41384)

- Agent startup failed on Solaris 10. The error generated was:

```
# ./mysql-monitor-agent start
Bad string
ERROR! /opt/mysql/enterprise/agent/etc/mysql-monitor-agent.ini has to have a
[mysql-proxy].pid-file setting
```

This was caused by unexpected behavior of the `tr` command. (Bug#41260)

- On the Query Analysis page, if a query was clicked the minimum displayed was greater than the average. (Bug#41259)
- In some circumstances the agent/proxy ran out of file descriptors, causing secondary failures. It could not recover from that state. The relevant part of the log file is shown here:

```
2008-11-27 11:11:00: (critical) last message repeated 2 times
2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
2008-11-27 11:11:00: (critical) job_collect_os.c:411: sigar_cpu_info_list_get() failed
2008-11-27 11:11:00: (critical) job_collect_os.c:445: sigar_cpu_list_get() failed
2008-11-27 11:11:30: (critical) network-socket.c:292: socket(127.0.0.1:3306) failed: Too many open files (24)
2008-11-27 11:11:30: (critical) proxy-plugin.c.1532: Cannot connect, all backends are down.
2008-11-27 11:20:22: (critical) last message repeated 4 times
2008-11-27 11:20:22: (critical) network-io.c:215:
curl_easy_perform('https://user:password@merlin-dashboard:443/heartbeat') failed:
SSL connection timeout (curl-error = 'Timeout was reached' (28), os-error = 'Connection refused' (111))
```

(Bug#41068)

- If an installation of Service Manager 2.0.0.7102 included a `backup` directory, due to a previous upgrade, and was upgraded using at least Service Manager 2.0.0.7103, then the installer displayed an error message and exited.

The error message displayed was:

```
There has been an error.
Error renaming /Applications/mysql/enterprise/monitor/apache-tomcat to
/Applications/mysql/enterprise/monitor/backup/apache-tomcat
The application will exit now
```

(Bug#40996)

- The Agent started without problems and connected to the master. But it was unable to perform a `SELECT` from the table `mysql.inventory` in order to obtain server information. (Bug#40933)
- Canonical Query Text for `Select -1` was displayed as `SELECT -?` instead of `SELECT ?` on the Query Analyzer tab. (Bug#40435)
- The agent installer for Solaris 8 x86 32-bit was missing. (Bug#40248)
- The startup scripts supplied with MySQL Network Monitoring and Advisory tool did not supply all of the LSB `init.d` script options required. A list of the required options can be found at the following website http://refspecs.freestandards.org/LSB_3.1.0/LSB-Core-generic/LSB-Core-generic/iniscriptact.html

The required options missing include `status` and `force-reload`. The option `status` is used by monitoring tools and cluster software such as Red Hat Cluster, to ensure that the service is still running. The `force-reload` option is an alias for restart. (Bug#29848)

- Multiple errors showed in the agent log after issuing a `SHOW INNODB STATUS` statement. The `InnoDB Buffer Poolgraph` also went blank. (Bug#27372)

D.4. Changes in MySQL Enterprise Monitor 2.0.4 (5th February 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.3.

Bugs fixed:

- The Service Agent did not log connections to, and disconnects from, the monitored database.

This meant it was not possible to check if the Agent was connected to the database by simply looking at the log file. ([Bug#42403](#))

- The Service Agent failed to create the `mysql.inventory` table. The logs displayed the following error message:

```
(critical) (share/mysql-proxy/quan.lua:711) [proxy] please add SELECT permissions for
this user on mysql.inventory to enable the QUAN feature, got Table 'mysql.inventory'
doesn't exist
```

This happened even though the Service Agent used the `root` account. ([Bug#42389](#))

- After installing Service Agent 2.0.4.7138 and then starting it with `--version`, the incorrect version was displayed. Although 2.0.4.7138 was installed, the Service Agent displayed the version number as 2.0.2.7138. ([Bug#42263](#))
- An invalid path was shown in the error message if the upgrade installer failed to find the previous install location.

The error message is shown below, note that the error message displays a different path to that provided by the user:

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

Installation directory [/home/mysql/mysql/enterprise/agent]: /var/lib/mysql/agent

Warning: The directory /home/mysql/mysql/enterprise/agent does not contain a
previous installation. Please select the right installation directory.
Press [Enter] to continue :
-----
Please specify the directory that contains the previous installation of the MySQL
Enterprise Monitor Agent
```

([Bug#42200](#))

- The agent password error in the Service Manager did not log the originating host, making it impossible to determine the agent that failed to log in:

```
ErrorJan 5, 2009 4:56:08 PM<?xml version='1.0'?><exceptions><error><![CDATA[E1702:
IncorrectPasswordException: [agent]]]></error></exceptions>
```

([Bug#42139](#))

- The Service Manager's JDBC connections did not have `sql_mode` set to include `NO_ENGINE_SUBSTITUTION`. This resulted in the failure of Data Definition Language (DDL) if the Service Manager was inadvertently pointed to an incorrect `mysqld` instance. ([Bug#42137](#))
- A number of Advisor rules had advice text that had not been translated into Japanese. The Advisors that contained untranslated rules included Performance, Schema and Security. ([Bug#42067](#))
- Service Agent configuration files had global read privileges on the filesystem. ([Bug#41794](#))
- When the log files rotated to the maximum allowed by the `log4j` configuration, the metadata contained in the `FileAppender` became out of synchronization due to a logic bug. This caused an assertion error on any subsequent request for the logs tab or data. ([Bug#41593](#))
- SNMP trap messages were sending 127.0.0.1 as the IP address, and there was no feature to allow the user to configure the IP address contained in the SNMP message, which would have been useful for troubleshooting. ([Bug#41361](#))
- On all available test systems, including SUSE Linux Enterprise Server (SLES) 9 and 10 x86-32, x86-64, IA64, and Ubuntu 6.10 x86-64, the agent memory grew incrementally within ~3 hours to ~25M.

The agents then switched to the 'red' condition on the dashboard and no more data was reported. The agent processes were still alive.

This was also present in MySQL Enterprise Monitor version 2.0.0.7111 on Suse 7.3, with the glibc.2.2 x86 agent.

There was no load on the monitored servers, and the problem also occurred when the “self-quan” was not configured. ([Bug#41244](#))

- Although the **MONITOR** tab loaded initially, after a 64-bit MySQL server running a 32-bit MySQL Monitor Agent was clicked, a Null Pointer Exception was generated. ([Bug#41164](#))
- The Service Agent startup configuration did not seem to work and did not generate a log file. ([Bug#40583](#))
- An error generated by a rule failed to clear.

When a rule was created with the `os:disk:fs_used` data item, if the instance was not a valid mount point then the Service Agent reported the error:

```
2008-08-11 17:57:00: (critical) disk-get failed for all: 2
```

Note the above instance was for “all”, and similar results occurred for instance “disk”.

The issue was that upon editing the rule, or even deleting the rule, the agent still showed the above error type. Restarting the agent and the monitoring service failed to remove the error. ([Bug#38709](#))

- If the “On Save send test trap” checkbox was checked when the SAVE button was clicked and the locale was set to Japanese, an error occurred. The orange error banner was displayed at the top of the page with the error message in Japanese. ([Bug#32069](#))

D.5. Changes in MySQL Enterprise Monitor 2.0.3 (23rd January 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.2.

Bugs fixed:

- The Service Agent failed to connect to the server, and no error message was displayed in the log file.

However, when the log verbosity level was set to `message`, the following message was recorded:

```
2009-01-12 13:34:43: (warning) agent_mysqlld.c:600: agent connecting to mysql-server
failed: mysql_real_connect(host = '127.0.0.1', port = 3306, socket = ''): Lost connection
to MySQL server at 'reading initial communication packet', system error: 0 (mysql-errno =
2013)
```

([Bug#42044](#))

- There was a mismatch between the contents of an SNMP Trap from MySQL Enterprise Monitor and the definition given in the file `MONITOR.MIB`. ([Bug#41912](#))
- If a copy was made of a standard rule, the resulting Wiki markup was incorrect, resulting in the display of user-interface text containing HTML markup. ([Bug#41375](#))
- Allowing the heat chart rules to be set to unscheduled caused the user interface to appear broken. ([Bug#41312](#))
- When a custom rule requiring disk information was created, for example:

```
[Expression]
%disk_reads% > THRESHOLD

[Thresholds]
Critical Alert: 9000
Warning Alert: 3000
Info Alert: 1000

[Variable Assignment]
variables: %disk_reads%
Data Item: os:disk:disk_reads
Instance: /
```

The following error was written to the file `mysql-service-agent.log`:

```
2007-09-05 17:11:00: (critical) disk-get failed for c0d0p1: 2
```

It therefore appeared that the Service Agent was not able to obtain the required information. ([Bug#30820](#))

D.6. Changes in MySQL Enterprise Monitor 2.0.2 (14th January 2009)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.1.

Bugs fixed:

- The COMPLETE SETUP button was missing from the **SETUP** page you were taken to after the current subscription expired. ([Bug#41685](#))
- If the `host-id` of the monitored instance and the `host-id` of the current agent did not match, the agent generated the following error message:

```
Please TRUNCATE TABLE mysql.inventory on this mysql-instance and restart the agent.
```

However, it did not suggest using `sql_log_bin = 0`. This is used for all other actions against this table so that they are not replicated to slaves, each of which has their own copy of this table. ([Bug#41673](#))

- The Agent did not start up when the monitored server had many databases and tables, and was under heavy load. This was because the `trigger_schema` query was taking too long on agent start up. ([Bug#41555](#))
- The Service Agent failed to install on Solaris 10 x86. The following error was generated:

```
Installing
0% _____ 50% _____ 100%
#####
Warning: Problem running post-install step.
Installation may not complete correctly
Error running /usr/local/mysqlagent/bin/mysql-monitor-agent
--defaults-file=/usr/local/mysqlagent/etc/mysql-monitor-agent.ini
--plugins=agent --agent-generate-uuid=true : 2008-12-12 13:06:02: (critical)
Conversion from character set '646' to 'UTF-8' is not supported
2008-12-12 13:06:02: (message) shutting down normally
```

([Bug#41445](#))

- The console/stdout appender remained in the `log4j` configuration, which meant that all the MySQL Enterprise Monitor server logs were duplicated to Catalina's stdout, and thus `catalina.out`, which was wasteful, especially as that file was not rotated or managed. ([Bug#41439](#))
- When creating new multiple user accounts, the first attempt worked fine. However, following attempts to create new users did not show the **QUERY ANALYZER OPTIONS** in the **CREATE USER** popup until the role field was changed. ([Bug#41430](#))
- When login privileges were required the Service Manager did not redirect the user to the login page. This resulted in error messages being displayed rather than simply redirecting the user to the login page. This problem typically occurred if it was necessary to restart Tomcat. ([Bug#41320](#))
- The monitor 2.0.0.7105 and 2.0.0.7122 Solaris Intel update installer quits unexpectedly. The installer exits from the GUI in the **BACKUP OF PREVIOUS INSTALLATION** screen, when OpenSolaris is running on top of Sun xVM.

The console output for both installer versions is given below:

```
shell> ./mysqlmonitor-2.0.0.7105-solaris-intel-update-installer.bin
X Error of failed request: BadMatch (invalid parameter attributes)
Major opcode of failed request: 73 (X_GetImage)
Serial number of failed request: 21161
Current serial number in output stream: 21161
```

```
shell> ./mysqlmonitor-2.0.0.7122-solaris-intel-update-installer.bin
X Error of failed request: BadMatch (invalid parameter attributes)
Major opcode of failed request: 73 (X_GetImage)
Serial number of failed request: 21148
Current serial number in output stream: 21148
```

([Bug#41315](#))

- The **RENAME GROUP** function failed if the new group name was different to the current one only in the case used; for example, if "Merlin" was changed to "MERLIN".

The error generated was:

```
U0105 This group name is already in use. Enter a different name.
```

(Bug#41266)

- The Agent returned an inventory list of all databases and tables. This information was not used by MySQL Enterprise Monitor, other than to populate the `inv_databases` and `inv_tables` tables. For large-scale deployments, where there were many databases and tables, this resulted in redundant XML messages being sent from the Agent to the Service Manager. (Bug#33150)

D.7. Changes in MySQL Enterprise Monitor 2.0.1 (15th December 2008)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 2.0.0.

Bugs fixed:

- Having an installation with over 420 master binlogs resulted in the `mysql::status` graph not being plotted. (Bug#41314)
- The MySQL Enterprise Monitor installer generated a segfault on RedHat Fedora 10:

```
shell> ./mysqlmonitor-2.0.0.7105-linux-x86-installer.bin
Segmentation fault
```

(Bug#41196)

D.8. Changes in MySQL Enterprise Monitor 2.0.0 (11th December 2008)

This section documents all changes and bug fixes that have been applied since the release of MySQL Enterprise Monitor, version 1.3.3.

Functionality added or changed:

- **Important Change:** The `server-name` configuration parameter is deprecated. For compatibility, during an upgrade, the information will be migrated to a `displayname` configuration parameter within the individual instance configuration files. This configuration parameter is provided only for compatibility, as display name information is now stored within the main repository. Support for `displayname` is also deprecated and will be removed in a future release.
- **Important Note:** The rules “32-Bit Binary Running on 64-Bit AMD Or Intel System” and “Key Buffer Size Greater Than 4 GB” occasionally do not evaluate correctly due to timing issues. This causes them to be displayed with the **SEVERITY** level of “Unknown”. This is a known issue and will be resolved in future versions of MySQL Enterprise Monitor.
- **Important Note:** When you start the Merlin 2.0 agent from the command line on Windows, you get the following error dialog:

```
"mysql-proxy.exe - Entry Point Not Found"
"The procedure entry point libiconv_set_relocation_prefix could not be located in the
dynamic link library iconv.dll"
```

If you click OK the agent works fine after that.

This only occurs when starting the agent from the command line, and only when there is another version of one of the DLLs that the agent uses somewhere on the current path. This error can be avoided by opening a command prompt, typing `SET PATH=` to clear the path, and then starting the agent.

- **Important Note:** If you are monitoring one instance of MySQL server (`mysqld`) and then upgrade that MySQL server, the correct version of the MySQL server is not displayed in the Dashboard. This is a known issue that will be fixed in future versions of MySQL Enterprise Monitor.
- The following has been added to the Tomcat `config.properties` properties file:

```
# max connections in the pool for the repository
default.maxActive=70
```

(Bug#40652)

- The dashboard could be used to change the agent password to one containing the @ character, or other special characters, which subsequently caused errors. To fix this problem, special characters in passwords are now prevented by the dashboard. The list of disallowed special characters can be found at the following location: http://en.wikipedia.org/wiki/Percent-encoding#Types_of_URI_characters (Bug#37172)
- The **QUERY ANALYSIS** page was missing the **REFRESH** dropdown control that the **MONITOR**, **EVENTS**, and **GRAPHS** pages had at the top. (Bug#36831)
- The User Interface only returned error strings, without any associated error codes. This meant that if the error string was in a language that the user did not understand, it would be very difficult to determine which error actually occurred.

The User Interface now supports error codes, as well as error strings. This change allows easier testing of multiple locales. (Bug#32131)

- The wording was changed on the fading popup subscription alert. The text “account” was changed to “subscription”. (Bug#31492)
- The alert thresholds for the Query Cache Advisor were changed:

	Information	Warning	Critical
From	95	85	75
To	60	50	40

(Bug#30969)

- The agent log file name has changed from `mysql-service-agent.log` to `mysql-monitor-agent.log`. The old log file will be retained during a upgrade install.

Bugs fixed:

- Starting a new agent against an instance that contained many databases broke up the initial discovery packet, causing collections such as CPU usage and their graphs to fail. See also Bug#33150 and Bug#41314. (Bug#41933)
- When altering an existing rule you had to add an empty string, " ", to any threshold level that was empty. Otherwise, the rule failed to run and the resulting exceptions caused the **EVENTS** page to be unusable due to duplicate key exceptions. (Bug#41310)
- After installing the 2.0.0.7119 Dashboard the following error was generated in the logs:

```
com.opensymphony.xwork2.util.OgnlValueStack Warning Dec 5, 2008 10:41:26 AM Caught
an Ognl exception while getting property titleAddition
```

(Bug#41252)

- The `dc_ng_long_now` table became very large partly due to an unused column `begin_time`. (Bug#41093)
- Although there was a unique constraint on the user name, it was not enforced during first-time setup. This resulted in a stack trace being produced, rather than a more user-friendly error message, if the same name was used for the `admin` and `agent` accounts. (Bug#40870)
- MySQL Enterprise Monitor server had stopped sending up/down SNMP traps. (Bug#40861)
- The Query Analyzer's **EXPLAIN QUERY** tab did not have pop up text or a link to the documentation regarding **SELECT** queries. (Bug#40841)
- When you tried to import a Trial-level advisor JAR using a Trial user account, one of the following error messages was generated:

```
U0009 The uploaded Advisor jar was invalid
```

```
U0161 Please import a Platinum level Advisor .jar to use with this Platinum level product key
```

(Bug#40834)

- Meta Info on the Dashboard did not display information for the meta data `os_description`. (Bug#40830)
- Attempting to create an alias of statements such as `COMMIT`, `ROLLBACK`, `BEGIN`, resulted in the error:

```
Can't find template commitTxn.st; group hierarchy is [HTMLFormatting]
org.apache.velocity.runtime.template.StringTemplateGroup.lookupTemplate(StringTemplateGroup.java:507)
```

(Bug#40736)

- Trying to upgrade from 2.0.0.7088 to 2.0.0.7092 failed as there was a missing file. When the update program `mysqlmonitor-2.0.0.7092-solaris-intel-update-installer.bin` was run, the file `/tmp/com/mysql/merlin/server/version.props` could not be found. (Bug#40692)

- On OS X with Java 1.5, Tomcat crashed on launch with the error:

```
Invalid memory access of location 00000007 eip=013df179
```

(Bug#40689)

- When the agent was installed using the command line installer and `--enableproxy 0` was specified, the installer should have removed `quan.lua` from the `agent-item-files` option in the INI file. (Bug#40551)
- The Agent could not connect to a database with the `hostname` set to `localhost`. Doing so resulted in the error:

```
(critical) the MySQL server could not be reached at socket '(null)', we will check in 10 seconds
```

(Bug#40530)

- The update installer for the 2.0 Monitor did not have an **IS SSL SUPPORT REQUIRED?** checkbox. Therefore, the appropriate SSL connector definition was commented out in the `conf/server.xml` file. (Bug#40414)
- The Service Manager installer did not uninstall or wipe out the previous installation if you answered “Yes” to the question: “The directory you selected already contains a MySQL installation. If you continue installation all data will be lost. Do you wish to continue?” (Bug#40410)
- If you unchecked the **ENABLE PROXY** checkbox on the **QUERY ANALYSIS CONFIGURATION** screen, the agent's INI file still contained proxy configuration data and was not commented out. (Bug#40272)
- As different queries were sent through the agent it used increasing amounts of memory. (Bug#40260)
- The Service Manager installer set the Java Virtual Machine option `HeapDumpPath` to `<install_root>\temp` on Windows and `/tmp` on other platforms. For consistency the `HeapDumpPath` directory should have been set to `<install_root>\tmp` on Windows. (Bug#40215)
- When using the command line agent setup program, a socket file was not accepted for the monitored instance. (Bug#40085)
- The language option when installing the MySQL Enterprise Monitor in Japanese using the command-line installer has been changed from `jp` to `ja`. (Bug#40082)
- When monitoring MySQL 5.1.24 and above, the user used by the agent to connect to the MySQL server for monitoring must have the `PROCESS` privilege. (Bug#40050)
- The check box option string “Is SSL support required?”, on the Tomcat Server Option dialog of the Monitor installation, was not correctly translated into Japanese. (Bug#39814)
- The base application directory for the MySQL Enterprise Dashboard has been updated from `http://localhost:18080/merlin` to `http://localhost:18080/`. (Bug#39403)
- If the Service Manager or the MySQL Server running the Repository crashed, they did not restart automatically. (Bug#39377)
- If the agent crashed, there was no watchdog, angel or keep-alive mechanism to restart the agent and keep it running. (Bug#39374)
- If the MySQL client libraries were located in a non-standard location, the agent 2.0.0.7042 installer failed with a “library not found” error. (Bug#39317)
- For the rule “Server-Enforced Data Integrity Checking Not Strict”, the Recommended Action did not display correctly. It displayed as `SET sql_mode=modes`, rather than `SET [GLOBAL|SESSION] sql_mode=modes`. (Bug#39261)
- A query that was issued through the proxy, and that had an auto-explain performed on that query, did not give the correct response to a subsequent query of `SELECT FOUND_ROWS()`. (Bug#39223)

- It was not possible to establish a connection with the Dashboard using the SSL protocol (https://). ([Bug#39198](#))
- When a 1.3 MySQL Enterprise Monitor installation with many rules scheduled was upgraded to 2.0, the upgraded installation was then found to have only the heat chart rules scheduled. ([Bug#39043](#))
- The agent installer did not allow a second agent to be installed on Windows. ([Bug#38976](#))
- The Dashboard incorrectly displayed that insufficient licenses were available, even though sufficient licenses had been purchased. ([Bug#38514](#))
- After running the MySQL Service Agent uninstall program, the file `/etc/init.d/mysql-service-agent` remained present on the server. ([Bug#38490](#))
- The notice fader continued to display English text after you changed the locale to Japanese. ([Bug#38460](#))
- The **SUBSCRIPTION EXPIRED** pop-up window referred to the “Global Preferences” page, instead of the “Global Settings” page. ([Bug#38358](#))
- An inappropriate time zone was used to parse user-entered date and time strings. ([Bug#38323](#))
- A sigar network stats error was generated on the Solaris platform:

```
# /opt/mysql/enterprise/agent/bin/mysql-service-agent --version
MySQL Service Agent - 1.2.0.7879, (glibc lib=2.8.5, headers=2.8.5)

SunOS mysqlprd01 5.10 Generic_127127-11 sun4v sparc SUNW,T5240

2008-07-21 10:07:24: (critical) sigar_net_interface_config_primary_get() failed: 6
2008-07-21 10:08:00: (critical) sigar_net_interface_config_primary_get() failed: 6

# /opt/mysql/enterprise/agent/bin/sigar-test-all >/tmp/test.txt
sigar_net_interface_stat_get(e1000g0:2) failed#
```

([Bug#38302](#))

- After deleting a server from the **SETTINGS, MANAGE SERVERS** tab, at the very bottom of the page the **MONITORING X INSTANCES ON X HOST** values did not reflect the deletion. ([Bug#38225](#))
- The MySQL Enterprise Monitor alert “INFO Alert - Users Can View All Databases On MySQL Server (v 1.5 *)” from the Security advisor was incorrect. This is because the default server behavior allows users to see databases for which they have privileges, not “all databases on server” as suggested by the alert. ([Bug#38052](#))
- The “Maximum Connection Limit Nearing Or Reached” advisor did not generate a new Critical Alert event when there was an open info success event. ([Bug#37816](#))
- The Linux IA64 installer appeared to crash. The installer appeared to crash on RH4_IA64 if called with option “--version”:

```
-----
<INSTALLER> --version
mysqlmonitorage(30704): unaligned access to 0x6000000000a8413c, ip=0x2000000003ddd5f0
mysqlmonitorage(30704): unaligned access to 0x6000000000a84144, ip=0x2000000003ddd5f1
mysqlmonitorage(30704): unaligned access to 0x6000000000a8414c, ip=0x2000000003ddd600
mysqlmonitorage(30704): unaligned access to 0x6000000000a84154, ip=0x2000000003ddd601
MySQL Enterprise Monitor Agent 0.7.0.1737 --- Built on 2008-06-25 19:31:53
-----
```

However, this warning is harmless and will not impact the operation of the agent. ([Bug#37496](#))

- If the log-level option in the agent configuration file was set to an unknown level by mistake, the `init.d` script appeared to enter an infinite loop. ([Bug#37108](#))
- Malformatted server meta information appeared on the Dashboard; **RAM** and **DISK SPACE** appeared under the **CPU** category. ([Bug#36740](#))
- A “rename” link incorrectly appeared next to the **UPGRADE** category on the **MANAGE RULES** page. ([Bug#36584](#))
- In the case where exceptions were passed through to the User Interface, the substituted arguments in the message contained developer-only information. ([Bug#36580](#))
- **AGENT VERSION, LAST MYSQL CONTACT, OS INFO, CPU INFO, and IP ADDRESSES** were all blank on the dashboard when the agent for the selected server was not functioning. ([Bug#36301](#))
- `mysql-monitor-agent` became confused by the `.DS_Store` files that are created on Mac OS X. ([Bug#36216](#))

- The rule “Key Buffer Size Greater Than 4 GB” incorrectly triggered the following alert:

`CRITICAL ALERT - KEY BUFFER SIZE GREATER THAN 4 GB`

However, on non-Windows systems, a key buffer size greater than 4 GB is supported. ([Bug#36143](#))

- Since the repository database for MySQL Enterprise Monitor uses InnoDB there was no way to reduce the size of the data files after an old log/event data purge operation. Further, the purge data operation executed once per day, and had no option to trigger the purge operation manually. ([Bug#35971](#))
- On the Graphs page, if all graphs were expanded, then the Time Display interval updated, the page was refreshed with the EXPAND ALL button displayed, even though all the graphs were already expanded. ([Bug#35917](#), [Bug#35133](#))
- The Meta Info area of the Monitor page incorrectly reported the operating system version number for the MySQL version. ([Bug#35836](#))
- The rule “XA Distributed Transaction Support Enabled For InnoDB” incorrectly sent a warning when the binary log was on. ([Bug#35786](#))
- On the Monitor page, the time displayed for `Last MySQL Contact` lagged behind that for `Last Agent Contact` by a large amount. ([Bug#35774](#))
- MySQL Enterprise Monitor did not update replication settings correctly. After a slave became the master, the Adviser still referred to it as a slave. ([Bug#35771](#))
- The Adviser email suggested using the `--log-queries-not-using-indexes` option. However, this option is not available in MySQL Server versions prior to 4.1. ([Bug#35770](#))
- Thumbnail graphs did not update properly after a time zone change. ([Bug#35756](#))
- If a system had a global `wait_timeout` lower than the general activity of the agent, the agent was disconnected. The monitored server then logged an error and incremented `Aborted_clients`. ([Bug#35648](#))
- Alerts fired after a blackout period based on data collections that happened during the blackout. ([Bug#35617](#))
- The translation of the ADD ROW button on the Rule Definition window was incorrect. ([Bug#35495](#))
- An uninstallation message asked about removing Apache files, even though Apache is no longer used. ([Bug#35154](#))
- After updating from a previous version to the latest 1.3 version, the **QUERY CACHE HAS SUB-OPTIMAL HIT RATE** was still displayed in English after setting the locale to Japanese. Note, the rule was translated correctly if the full installer was used. ([Bug#35134](#))
- The MySQL Enterprise Monitor uninstall dialog box had missing text when using the Japanese locale. ([Bug#34982](#))
- Running the installer with the `--help` option caused an incorrect message to be displayed. ([Bug#34200](#))
- When using the unattended `unInstall` script on Linux together with the option `--env deleteUserData=yes` the correct warning text was displayed. However, this text should not be displayed in unattended mode. Further, the option `--env deleteUserData=yes` was not displayed by the `--help` output. ([Bug#34071](#))
- Platinum Unlimited customers sometimes received a warning stating incorrectly that their subscription supported zero hosts. ([Bug#34010](#))
- Clicking on the resolution notes link for a closed event on the events tab showed incorrect behaviour. The popup initially showed the resolution notes, but when the resolution tab was clicked the notes disappeared and were replaced by an edit box. ([Bug#33935](#))
- The status on the product information page was not translated when the user locale was set to Japanese. ([Bug#32785](#))
- When the locale was set to Japanese, the date picker still had English month titles. ([Bug#32741](#))
- Flashing display of a pop-up used while saving outgoing email settings was caused by problematic initial placement calculations. ([Bug#32579](#))
- AIX 5.2 Agent did not work on AIX 5.3. ([Bug#32414](#))
- When the First Time Setup program was run it did not prompt the user to allow importing an Advisor bundle. ([Bug#32199](#))
- Agent on MacOSX did not read IP addresses for network interfaces correctly, so the monitor displayed empty host IP addresses. ([Bug#32188](#))

- HTML code in queries was not escaped when reporting replication errors, causing the code to be rendered into the page. (Bug#32186)
- The First Run pop-up defaulted to English rather than to the locale set in the browser. (Bug#32129)
- The error dialog box flashed in the upper left corner before being positioned in the center of the screen. This error dialog box now opens in the center of the screen. (Bug#32068)
- The Events list did not take into account Daylight Time and Standard Time when listing events that happened during 1:00am-1:59am. An event that occurred at 1:10am Standard Time was listed before an event that occurred 50 minutes before it at 1:20am Daylight Time. (Bug#32016)
- The pop-up for editing log levels failed to load due to bad instantiation data. (Bug#32013)
- During the repeated hour of Daylight Savings Time (when 2am turns back into 1am), the graphs were not drawing data. Instead, there was a straight line from the point at 1:00 to the second 1:00, which is what happens if there is no data. The repository did, however, have data for this hour. (Bug#31997)
- Only US English was supported for a locale setting. Other English variants are now available for the `locale` setting on the [General Settings](#) or the [User Preferences](#) pages. (Bug#31801)
- If the user locale was changed the graph cache would continue to display the graph in the last locale until it timed out. (Bug#31680)
- No init script was installed for the MySQL Network Monitoring Service Manager, and so it did not restart automatically on reboot. (Bug#31676)
- The graph's displayed time was not the local time of the Dashboard corresponding to the requested time on the monitored server. (Bug#31656)
- Saving a rule with a name that already existed resulted in a stack trace in the window, instead of a more user-friendly error message. (Bug#30925)
- The `network.mysql.com` error messages were remapped thereby causing confusion. For example, the following error message:

```
E9000: MySQL ENTERPRISE CUSTOMER CENTER IS HAVING DIFFICULTIES FETCHING YOUR CONTRACT INFORMATION.  
PLEASE CONTACT ENTERPRISE-FEEDBACK@MYSQL.COM FOR ASSISTANCE.
```

Was remapped to:

```
UNABLE TO CONNECT TO VERIFY CREDENTIALS (Bug#30873)
```
- A newly added server showed as “down” in the user interface, and could potentially have sent a false alarm notification. (Bug#30735)
- The information on the [ADVISORS, CHECK FOR UPDATES](#) page did not accurately reflect how many rules and graphs were actually in the database and available to the user. (Bug#29623)
- The agent did not process `SIGHUP`. (Bug#29380)
- Monitor did not have a facility to stop or downgrade an agent collection frequency. (Bug#28589)
- After an agent installation was updated from 1.0.1.4391 to 1.1.0.4899, the version in the [ADD/REMOVE PROGRAMS](#) menu was incorrectly displayed as 1.0.1.4391, even though the update was successful and the file version of `agent.exe` was correctly displayed as 1.1.0.4899. (Bug#27447)
- When viewing the **RESULTS** of an **EVENT** in the **EVENTS** tab of the **DASHBOARD**, the **NOTIFICATIONS** section did not reflect the **NOTIFICATIONS** settings at the time the **EVENT** was triggered, but rather the **NOTIFICATIONS** settings at the time the **EVENT RESULTS** were viewed. (Bug#26349)
- When the Service Agent was remotely monitoring a MySQL server it incorrectly reported that it could collect operating system information. (Bug#22497)
- The Account Without Password advisor did not report all users who were without a password, it only reported one. (Bug#15165)

Appendix E. The Data Collection Items Used to Create Rules

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 6.3, “Editing Built-in Rules”](#) and [Section 6.4.2, “Overview of Rule Creation”](#).

Table E.1. Data Collection Items

Namespace	Type	Attribute	Type	Description
monitor	resourceBundle	subscription	string	
monitor	resourceBundle	version	string	
mysql	account_old_password	user	string	
mysql	active_count	active_count	long	
mysql	Agent	agent.reachable	string	
mysql	Agent	host_id	string	
mysql	Agent	name	string	
mysql	Agent	version	string	
mysql	anonymous_user	user_count	long	
mysql	avail_count	avail_count	long	
mysql	broad_host_specifier	user	string	
mysql	column	Default	string	The default value of the column
mysql	column	Extra	string	Any additional information about the column
mysql	column	Field	string	The name of the column
mysql	column	Key	string	Whether the column is indexed
mysql	column	Null	string	Whether NULL is allowed
mysql	column	Type	string	The data type of the column
mysql	committed_count	committed_count	long	
mysql	Database	Database	string	
mysql	Database	name	string	
mysql	Explain	extra	string	
mysql	Explain	id	long	
mysql	Explain	key	string	
mysql	Explain	key_len	string	
mysql	Explain	possible_keys	string	
mysql	Explain	ref	string	
mysql	Explain	rows	long	
mysql	Explain	select_type	string	
mysql	Explain	table	string	
mysql	Explain	type	string	
mysql	falcon_database_io	logical_reads	long	
mysql	falcon_database_io	physical_reads	long	
mysql	falcon_record_cache_summary	total_recordcache_mb	double	
mysql	falcon_record_cache_summary	total_space_mb	double	

Namespace	Type	Attribute	Type	Description
	ary			
mysql	fal-con_system_memory_summary	free_system_mb	double	
mysql	fal-con_system_memory_summary	total_system_mb	double	
mysql	fal-con_transaction_summary	committed_txns	long	
mysql	fal-con_transaction_summary	rolled_back_txns	long	
mysql	global_privileges	user	string	
mysql	grant_privileges	user_spec	string	
mysql	inappropriate_slave_privileges	user	string	
mysql	index	Cardinality	long	An estimate of the number of unique values in the index
mysql	index	Collation	string	How the column is sorted in the index
mysql	index	Column_name	string	The column name
mysql	index	Comment	string	Remarks about the index
mysql	index	Index_type	string	The index method used (BTREE, FULLTEXT, HASH, RTREE)
mysql	index	Key_name	string	The name of the index
mysql	index	Non_unique	long	Whether the index can contain duplicates
mysql	index	Null	string	Whether the column may contain NULL values
mysql	index	Packed	string	Whether the key is packed
mysql	index	Seq_in_index	long	The column sequence number in the index, starting with 1
mysql	index	Sub_part	string	The number of indexed characters if the column is only partly indexed
mysql	index	Table	string	The name of the table associated with the index
mysql	innodbstatus	innodb_bp_add_alloc	long	The total memory allocated for the additional buffer pool measured in bytes
mysql	innodbstatus	innodb_bp_created_per_sec	double	The number of buffer pool pages created per second
mysql	innodbstatus	innodb_bp_db_pages	long	The current number of buffer pool pages
mysql	innodbstatus	innodb_bp_free_buffers	long	The current number of free buffer pool pages
mysql	innodbstatus	innodb_bp_hit_rate	long	The buffer pool hit rate
mysql	innodbstatus	innodb_bp_modified_pages	long	The current number of pages modified
mysql	innodbstatus	innodb_bp_pages_created	long_counter	The total number of pages created
mysql	innodbstatus	innodb_bp_pages_read	long_counter	The total number of pages read
mysql	innodbstatus	innodb_bp_pages_written	long_counter	The total number of pages written
mysql	innodbstatus	innodb_bp_pending_pages	long	The number of pending page writes
mysql	innodbstatus	innodb_bp_pending_writes_flush_list	long	The number of pages to be flushed during checkpointing
mysql	innodbstatus	innodb_bp_pending_writes_lru	long	The number of old dirty pages to be written from the bottom of the LRU list

Namespace	Type	Attribute	Type	Description
mysql	innodbstatus	innodb_bp_pending_writes_single_page	long	The number of pending independent page writes
mysql	innodbstatus	innodb_bp_reads_per_sec	double	The number of buffer pool reads per second
mysql	innodbstatus	innodb_bp_size	long	The total buffer pool size in bytes
mysql	innodbstatus	innodb_bp_total_alloc	long	The total memory allocated for the buffer pool
mysql	innodbstatus	innodb_bp_written_per_sec	double	The number of buffer pool pages written per second
mysql	innodbstatus	innodb_bytes_per_read	long	The number of bytes per read
mysql	innodbstatus	innodb_datetime	string	The date and time the SHOW ENGINE INNODB STATUS snapshot was taken
mysql	innodbstatus	innodb_hash_node_heap	long	Number of buffer pool pages reserved for the Adaptive Hash Index
mysql	innodbstatus	innodb_hash_searches_per_sec	double	The number of hash searches per second
mysql	innodbstatus	innodb_hash_table_size	long	The size of the hash table
mysql	innodbstatus	innodb_hash_used_cells	long	Number of buffer pool pages used for the Adaptive Hash Index
mysql	innodbstatus	innodb_ibuf_inserts	long_counter	The number of insert buffer inserts
mysql	innodbstatus	innodb_ibuf_merged_recs	long_counter	The number of insert buffer merged records
mysql	innodbstatus	innodb_ibuf_merges	long_counter	The number of insert buffer merges
mysql	innodbstatus	innodb_io_ibuf_logs	long	The number of pending log I/Os
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
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

Namespace	Type	Attribute	Type	Description
mysql	innodbstatus	innodb_log_sequence_lsn	long	The log sequence number
mysql	innodbstatus	innodb_non_hash_searches_per_sec	double	The number of non-adaptive hash index searches
mysql	innodbstatus	innodb_per_sec_avg	double	The number of seconds the averages for SHOW INNODB STATUS were calculated from
mysql	innodbstatus	innodb_reads_per_sec	double	The number of reads per second
mysql	innodbstatus	innodb_row_queries_inside	long	The number of queries executing inside InnoDB
mysql	innodbstatus	innodb_row_queries_queue	long	The number of queries in the queue, waiting to enter InnoDB
mysql	innodbstatus	innodb_row_state	string	The current state of the main InnoDB thread
mysql	innodbstatus	innodb_rows_deleted	long_counter	The number of rows deleted from InnoDB tables
mysql	innodbstatus	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
mysql	innodbstatus	innodb_sem_shared_spins	long_counter	The number of shared (read) semaphore spin waits within the syn array
mysql	innodbstatus	innodb_trx_history_list_length	long	The transaction history list length
mysql	innodbstatus	innodb_trx_id_counter1	long	The transaction counter roll over variable
mysql	innodbstatus	innodb_trx_id_counter2	long	The main transaction count
mysql	innodbstatus	innodb_trx_purge_done_trx2	long	The transaction count that is already purged
mysql	innodbstatus	innodb_trx_total_lock_structs	long	
mysql	innodbstatus	innodb_writes_per_sec	double	The number of InnoDB writes per second
mysql	masterlogs	File_size	long	The size of a binary log file

Namespace	Type	Attribute	Type	Description
mysql	masterlogs	filecount	long	The number of binary log files
mysql	masterlogs	filesizesum	long	The total size of all the binary log files
mysql	masterlogs	Log_name	string	The name of the binary log file
mysql	MasterStatus	Binlog_Do_DB	string	
mysql	MasterStatus	Binlog_Ignore_DB	string	
mysql	MasterStatus	File	string	
mysql	MasterStatus	Position	long	
mysql	no_password	user	string	
mysql	no_root_password	no_password	long	
mysql	num_waiting_txns	num_waiting_txns	long	
mysql	prepared_count	prepared_count	long	
mysql	privileges_on_all_dbs	user_spec	string	
mysql	processlist	Command	string	The type of command the thread is executing
mysql	processlist	db	string	The default database, if one is selected
mysql	processlist	Host	string	The host name of the client issuing the statement
mysql	processlist	Id	long	The connection identifier
mysql	processlist	Info	string	The statement that the thread is executing
mysql	processlist	State	string	An action, event, or state that indicates what the thread is doing
mysql	processlist	Time	long	The time in seconds that the thread has been in its current state
mysql	processlist	User	string	The MySQL user who issued the statement
mysql	repl_slave_privileges	repl_slaves	long	
mysql	rolledback_count	rolledback_count	long	
mysql	root_account_exists	root_account	long	
mysql	root_remote_login	remote_login	long	
mysql	Server	agent.reachable	string	Whether the agent is reachable or not
mysql	Server	blackout	string	
mysql	Server	displayname	string	The display name of the server in the Dashboard
mysql	Server	registration-complete	string	
mysql	Server	repl.group	long	
mysql	Server	repl.groupName	string	
mysql	Server	server.connected	long	Whether the server is connected
mysql	Server	server.last_errno	string	
mysql	Server	server.last_error	string	The last MySQL server error message
mysql	Server	server.reachable	long	Whether the server is reachable
mysql	Server	server.version_numeric	long	The MySQL server version number
mysql	Server	Time	long	
mysql	Server	transport	string	
mysql	Server	visible.displayname	string	
mysql	server_admin_privileges	user_spec	string	
mysql	slavestatus	Binlog_Do_DB	string	
mysql	slavestatus	Binlog_Ignore_DB	string	
mysql	slavestatus	Connect_Retry	long	
mysql	slavestatus	Exec_Master_Log_Pos	long	

Namespace	Type	Attribute	Type	Description
mysql	slavestatus	File	string	
mysql	slavestatus	Last_Errno	long	
mysql	slavestatus	Last_Error	string	
mysql	slavestatus	Master_Host	string	
mysql	slavestatus	Master_ip	string	
mysql	slavestatus	Master_Log_File	string	
mysql	slavestatus	Master_Port	long	
mysql	slavestatus	Master_SSL_Allowed	string	
mysql	slavestatus	Master_SSL_CA_File	string	
mysql	slavestatus	Master_SSL_CA_Path	string	
mysql	slavestatus	Master_SSL_Cert	string	
mysql	slavestatus	Master_SSL_Cipher	string	
mysql	slavestatus	Master_SSL_Key	string	
mysql	slavestatus	Master_User	string	
mysql	slavestatus	Position	long	
mysql	slavestatus	Read_Master_Log_Pos	long	
mysql	slavestatus	Relay_Log_File	string	
mysql	slavestatus	Relay_Log_Pos	long	
mysql	slavestatus	Relay_Log_Space	long	
mysql	slavestatus	Relay_Master_Log_File	string	
mysql	slavestatus	Replicate_Do_DB	string	
mysql	slavestatus	Replicate_Do_Table	string	
mysql	slavestatus	Replicate_Ignore_DB	string	
mysql	slavestatus	Replicate_Ignore_Table	string	
mysql	slavestatus	Replicate_Wild_Do_Table	string	
mysql	slavestatus	Replicate_Wild_Ignore_Table	string	
mysql	slavestatus	Seconds_Behind_Master	long	
mysql	slavestatus	Skip_Counter	long	
mysql	slavestatus	Slave_IO_Running	string	
mysql	slavestatus	Slave_IO_State	string	
mysql	slavestatus	Slave_SQL_Running	string	
mysql	slavestatus	Until_Condition	string	
mysql	slavestatus	Until_Log_File	string	
mysql	slavestatus	Until_Log_Pos	long	
mysql	sp_with_select_star	routine	string	
mysql	Statement	bytes	long	
mysql	Statement	comment	string	
mysql	Statement	connection_id	long	
mysql	Statement	exec_time	long	
mysql	Statement	host_from	string	
mysql	Statement	host_to	string	
mysql	Statement	max_bytes	long	
mysql	Statement	rows	long	
mysql	Statement	text	string	
mysql	Statement	user	string	
mysql	StatementAnalysisSupport	capture_examples	string	
mysql	StatementAnalysisSupport	capture_explain	string	

Namespace	Type	Attribute	Type	Description
mysql	StatementAnalysisSupport	enabled	string	
mysql	StatementAnalysisSupport	frequency	string	
mysql	StatementSummary	bytes	long	
mysql	StatementSummary	bytes_rel	long	
mysql	StatementSummary	count	long	
mysql	StatementSummary	count_rel	long	
mysql	StatementSummary	database	string	
mysql	StatementSummary	exec_time	long	
mysql	StatementSummary	exec_time_rel	long	
mysql	StatementSummary	max_bytes	long	
mysql	StatementSummary	max_exec_time	long	
mysql	StatementSummary	max_rows	long	
mysql	StatementSummary	min_bytes	long	
mysql	StatementSummary	min_exec_time	long	
mysql	StatementSummary	min_rows	long	
mysql	StatementSummary	query_type	string	
mysql	StatementSummary	rows	long	
mysql	StatementSummary	rows_rel	long	
mysql	StatementSummary	text	string	
mysql	StatementSummary	text_hash	string	
mysql	status	Aborted_clients	long_counter	Networking.Overview
mysql	status	Aborted_connects	long_counter	The number of failed attempts to connect to the MySQL server
mysql	status	Binlog_cache_disk_use	long	General.Logging
mysql	status	Binlog_cache_use	long	The number of transactions that used the temporary binary log cache
mysql	status	Bytes_received	long_counter	The number of bytes received from all clients
mysql	status	Bytes_sent	long_counter	The number of bytes sent to all clients
mysql	status	Com_admin_commands	long_counter	Count of admin commands
mysql	status	Com_alter_db	long_counter	Count of ALTER DATABASE statements
mysql	status	Com_alter_db_upgrade	long_counter	
mysql	status	Com_alter_event	long_counter	
mysql	status	Com_alter_function	long_counter	
mysql	status	Com_alter_procedure	long_counter	
mysql	status	Com_alter_server	long_counter	
mysql	status	Com_alter_table	long_counter	Count of ALTER TABLE statements
mysql	status	Com_alter_tablespace	long_counter	
mysql	status	Com_analyze	long_counter	Count of ANALYZE statements
mysql	status	Com_assign_to_keycache	long_counter	
mysql	status	Com_backup_table	long_counter	Count of BACKUP TABLE statements
mysql	status	Com_begin	long_counter	Count of BEGIN statements
mysql	status	Com_binlog	long_counter	
mysql	status	Com_call_procedure	long_counter	Number of calls to stored procedures
mysql	status	Com_change_db	long_counter	Count of CHANGE DATABASE statements
mysql	status	Com_change_master	long_counter	Count of CHANGE MASTER state-

Namespace	Type	Attribute	Type	Description
				ments
mysql	status	Com_check	long_counter	Count of CHECK statements
mysql	status	Com_checksum	long_counter	Count of CHECKSUM statements
mysql	status	Com_commit	long_counter	Count of COMMIT statements
mysql	status	Com_create_db	long_counter	Count of CREATE DATABASE statements
mysql	status	Com_create_event	long_counter	
mysql	status	Com_create_function	long_counter	Count of CREATE FUNCTION statements
mysql	status	Com_create_index	long_counter	Count of CREATE INDEX statements
mysql	status	Com_create_procedure	long_counter	
mysql	status	Com_create_server	long_counter	
mysql	status	Com_create_table	long_counter	Count of CREATE TABLE statements
mysql	status	Com_create_trigger	long_counter	
mysql	status	Com_create_udf	long_counter	
mysql	status	Com_create_user	long_counter	Count of CREATE USER statements
mysql	status	Com_create_view	long_counter	
mysql	status	Com_dealloc_sql	long_counter	Count of DEALLOCATE PREPARE statements
mysql	status	Com_delete	long_counter	Count of DELETE statements
mysql	status	Com_delete_multi	long_counter	Count of multiple table DELETE statements
mysql	status	Com_do	long_counter	Count of DO statements
mysql	status	Com_drop_db	long_counter	Count of DROP DATABASE statements
mysql	status	Com_drop_event	long_counter	
mysql	status	Com_drop_function	long_counter	Count of DROP FUNCTION statements
mysql	status	Com_drop_index	long_counter	Count of DROP INDEX statements
mysql	status	Com_drop_procedure	long_counter	
mysql	status	Com_drop_server	long_counter	
mysql	status	Com_drop_table	long_counter	Count of DROP TABLE statements
mysql	status	Com_drop_trigger	long_counter	
mysql	status	Com_drop_user	long_counter	Count of DROP USER statements
mysql	status	Com_drop_view	long_counter	
mysql	status	Com_empty_query	long_counter	
mysql	status	Com_execute_sql	long_counter	Count of EXECUTE statements
mysql	status	Com_flush	long_counter	Count of FLUSH statements
mysql	status	Com_grant	long_counter	Count of GRANT statements
mysql	status	Com_ha_close	long_counter	Count of HANDLER CLOSE statements
mysql	status	Com_ha_open	long_counter	Count of HANDLER OPEN statements
mysql	status	Com_ha_read	long_counter	Count of HANDLER READ statements
mysql	status	Com_help	long_counter	Count of HELP statements
mysql	status	Com_insert	long_counter	Count of INSERT statements
mysql	status	Com_insert_select	long_counter	Count of INSERT SELECT statements
mysql	status	Com_install_plugin	long_counter	
mysql	status	Com_kill	long_counter	Count of KILL statements
mysql	status	Com_load	long_counter	Count of LOAD statements

Namespace	Type	Attribute	Type	Description
mysql	status	Com_load_master_data	long_counter	Count of LOAD MASTER DATA statements
mysql	status	Com_load_master_table	long_counter	Count of LOAD MASTER TABLE statements
mysql	status	Com_lock_tables	long_counter	Count of LOCK TABLES statements
mysql	status	Com_optimize	long_counter	Count of OPTIMIZE statements
mysql	status	Com_preload_keys	long_counter	Count of PRELOAD KEYS statements
mysql	status	Com_prepare_sql	long_counter	Count of PREPARE statements
mysql	status	Com_purge	long_counter	Count of PURGE statements
mysql	status	Com_purge_before_date	long_counter	Count of PURGE BEFORE DATE statements
mysql	status	Com_release_savepoint	long_counter	
mysql	status	Com_rename_table	long_counter	Count of RENAME TABLE statements
mysql	status	Com_rename_user	long_counter	
mysql	status	Com_repair	long_counter	Count of REPAIR statements
mysql	status	Com_replace	long_counter	Count of REPLACE statements
mysql	status	Com_replace_select	long_counter	Count of REPLACE SELECT statements
mysql	status	Com_reset	long_counter	Count of RESET statements
mysql	status	Com_restore_table	long_counter	Count of RESTORE TABLE statements
mysql	status	Com_revoke	long_counter	Count of REVOKE statements
mysql	status	Com_revoke_all	long_counter	Count of REVOKE ALL statements
mysql	status	Com_rollback	long_counter	Count of ROLLBACK statements
mysql	status	Com_rollback_to_savepoint	long_counter	
mysql	status	Com_savepoint	long_counter	Count of SAVEPOINT statements
mysql	status	Com_select	long_counter	Count of SELECT statements
mysql	status	Com_set_option	long_counter	Count of SET OPTION statements
mysql	status	Com_show_authors	long_counter	
mysql	status	Com_show_binlog_events	long_counter	Count of SHOW BINLOG EVENTS statements
mysql	status	Com_show_binlogs	long_counter	Count of SHOW BINLOGS statements
mysql	status	Com_show_charsets	long_counter	Count of SHOW CHARSETS statements
mysql	status	Com_show_collations	long_counter	Count of SHOW COLLATIONS statements
mysql	status	Com_show_column_types	long_counter	Count of SHOW COLUMN TYPES statements
mysql	status	Com_show_contributors	long_counter	
mysql	status	Com_show_create_db	long_counter	Count of SHOW CREATE DATABASE statements
mysql	status	Com_show_create_event	long_counter	
mysql	status	Com_show_create_func	long_counter	
mysql	status	Com_show_create_proc	long_counter	
mysql	status	Com_show_create_table	long_counter	Count of SHOW CREATE TABLE statements
mysql	status	Com_show_create_trigger	long_counter	
mysql	status	Com_show_databases	long_counter	Count of SHOW DATABASES statements
mysql	status	Com_show_engine_logs	long_counter	
mysql	status	Com_show_engine_mutex	long_counter	

Namespace	Type	Attribute	Type	Description
mysql	status	Com_show_engine_status	long_counter	
mysql	status	Com_show_errors	long_counter	Count of SHOW ERRORS statements
mysql	status	Com_show_events	long_counter	
mysql	status	Com_show_fields	long_counter	Count of SHOW FIELDS statements
mysql	status	Com_show_function_status	long_counter	
mysql	status	Com_show_grants	long_counter	Count of SHOW GRANTS statements
mysql	status	Com_show_innodb_status	long_counter	Count of SHOW INNODB STATUS statements
mysql	status	Com_show_keys	long_counter	Count of SHOW KEYS statements
mysql	status	Com_show_logs	long_counter	Count of SHOW LOGS statements
mysql	status	Com_show_master_status	long_counter	Count of SHOW MASTER STATUS statements
mysql	status	Com_show_ndb_status	long_counter	Count of SHOW NDB STATUS statements
mysql	status	Com_show_new_master	long_counter	Count of SHOW NEW MASTER statements
mysql	status	Com_show_open_tables	long_counter	Count of SHOW OPEN TABLES statements
mysql	status	Com_show_plugins	long_counter	
mysql	status	Com_show_privileges	long_counter	Count of SHOW PRIVILEGES statements
mysql	status	Com_show_procedure_status	long_counter	
mysql	status	Com_show_processlist	long_counter	Count of SHOW PROCESSLIST statements
mysql	status	Com_show_profile	long_counter	
mysql	status	Com_show_profiles	long_counter	
mysql	status	Com_show_slave_hosts	long_counter	Count of SHOW SLAVE HOSTS statements
mysql	status	Com_show_slave_status	long_counter	Count of SHOW SLAVE STATUS statements
mysql	status	Com_show_status	long_counter	Count of SHOW STATUS statements
mysql	status	Com_show_storage_engines	long_counter	Count of SHOW STORAGE ENGINES statements
mysql	status	Com_show_table_status	long_counter	
mysql	status	Com_show_tables	long_counter	Count of SHOW TABLES statements
mysql	status	Com_show_triggers	long_counter	Count of SHOW TRIGGERS statements
mysql	status	Com_show_variables	long_counter	Count of SHOW VARIABLES statements
mysql	status	Com_show_warnings	long_counter	Count of SHOW WARNINGS statements
mysql	status	Com_slave_start	long_counter	Count of SLAVE START statements
mysql	status	Com_slave_stop	long_counter	Count of SLAVE STOP statements
mysql	status	Com_stmt_close	long_counter	Count of STATEMENT CLOSE statements
mysql	status	Com_stmt_execute	long_counter	Count of STATEMENT EXECUTE statements
mysql	status	Com_stmt_fetch	long_counter	Count of STATEMENT FETCH statements
mysql	status	Com_stmt_prepare	long_counter	Count of STATEMENT PREPARE statements
mysql	status	Com_stmt_reprepare	long_counter	
mysql	status	Com_stmt_reset	long_counter	Count of STATEMENT RESET statements

Namespace	Type	Attribute	Type	Description
				ments
mysql	status	Com_stmt_send_long_data	long_counter	Count of STATEMENT SEND LONG DATA statements
mysql	status	Com_truncate	long_counter	Count of TRUNCATE statements
mysql	status	Com_uninstall_plugin	long_counter	
mysql	status	Com_unlock_tables	long_counter	Count of UNLOCK TABLES statements
mysql	status	Com_update	long_counter	Count of UPDATE statements
mysql	status	Com_update_multi	long_counter	Count of multiple UPDATE statements
mysql	status	Com_xa_commit	long_counter	Count of XA COMMIT statements
mysql	status	Com_xa_end	long_counter	Count of XA END statements
mysql	status	Com_xa_prepare	long_counter	Count of XA PREPARE statements
mysql	status	Com_xa_recover	long_counter	Count of XA RECOVER statements
mysql	status	Com_xa_rollback	long_counter	Count of XA ROLLBACK statements
mysql	status	Com_xa_start	long_counter	Count of XA START statements
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

Namespace	Type	Attribute	Type	Description
mysql	status	Handler_savepoint	long_counter	The number of requests for a storage engine to place a savepoint
mysql	status	Handler_savepoint_rollback	long_counter	The number of requests for a storage engine to roll back to a savepoint
mysql	status	Handler_update	long_counter	The number of requests to update a row in a table
mysql	status	Handler_write	long_counter	The number of requests to insert a row in a table
mysql	status	Innodb_buffer_pool_pages_data	long	The number of pages containing data (dirty or clean)
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_random	long_counter	The number of random read-aheads initiated by InnoDB
mysql	status	Innodb_buffer_pool_read_ahead_seq	long_counter	The number of sequential read-aheads initiated by InnoDB
mysql	status	Innodb_buffer_pool_read_requests	long_counter	The number of logical read requests InnoDB has done
mysql	status	Innodb_buffer_pool_reads	long_counter	Performance.InnoDB
mysql	status	Innodb_buffer_pool_wait_free	long_counter	Counts number of waits for pages to be flushed
mysql	status	Innodb_buffer_pool_write_requests	long_counter	The number of writes done to the buffer pool
mysql	status	Innodb_data_fsyncs	long_counter	The number of fsync() operations so far
mysql	status	Innodb_data_pending_fsyncs	long	The current number of pending fsync() operations
mysql	status	Innodb_data_pending_reads	long	The current number of pending reads
mysql	status	Innodb_data_pending_writes	long	The number of pending writes
mysql	status	Innodb_data_read	long_counter	The amount of data read so far, in bytes
mysql	status	Innodb_data_reads	long_counter	The total number of data reads
mysql	status	Innodb_data_writes	long_counter	The total number of data writes
mysql	status	Innodb_data_written	long_counter	The amount of data written in bytes
mysql	status	Innodb_dblwr_pages_written	long_counter	The number of doublewrite pages that have been written
mysql	status	Innodb_dblwr_writes	long_counter	The number of doublewrite operations 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

Namespace	Type	Attribute	Type	Description
				log
mysql	status	Innodb_os_log_fsyncs	long_counter	The number of fsync() writes done to the log file
mysql	status	Innodb_os_log_pending_fsyncs	long	The number of pending log file fsync() operations
mysql	status	Innodb_os_log_pending_writes	long	The number of pending log file writes
mysql	status	Innodb_os_log_written	long_counter	The number of bytes written to the log file
mysql	status	Innodb_page_size	long	The compiled-in InnoDB page size
mysql	status	Innodb_pages_created	long_counter	The number of pages created
mysql	status	Innodb_pages_read	long_counter	The number of pages read
mysql	status	Innodb_pages_written	long_counter	The number of pages written
mysql	status	Innodb_row_lock_current_waits	long	The number of row locks currently being waited for
mysql	status	Innodb_row_lock_time	long_counter	The total time spent in acquiring row locks, in milliseconds
mysql	status	Innodb_row_lock_time_avg	long	The average time to acquire a row lock, in milliseconds
mysql	status	Innodb_row_lock_time_max	long	The maximum time to acquire a row lock, in milliseconds
mysql	status	Innodb_row_lock_waits	long_counter	The number of times a row lock had to be waited for
mysql	status	Innodb_rows_deleted	long_counter	The number of rows deleted from InnoDB tables
mysql	status	Innodb_rows_inserted	long_counter	The number of rows inserted into InnoDB tables
mysql	status	Innodb_rows_read	long_counter	The number of rows read from InnoDB tables
mysql	status	Innodb_rows_updated	long_counter	The number of rows updated in InnoDB tables
mysql	status	Key_blocks_not_flushed	long	Performance.Keys
mysql	status	Key_blocks_unused	long	The number of unused blocks in the key cache
mysql	status	Key_blocks_used	long	The number of used blocks in the key cache
mysql	status	Key_read_requests	long_counter	The number of requests to read a key block from the cache
mysql	status	Key_reads	long_counter	The number of physical reads of a key block from disk
mysql	status	Key_write_requests	long_counter	The number of requests to write a key block to the cache
mysql	status	Key_writes	long_counter	The number of physical writes of a key block to disk
mysql	status	Last_query_cost	long	Performance.Cost
mysql	status	Max_used_connections	long	Networking.Overview
mysql	status	Not_flushed_delayed_rows	long	The number of rows waiting to be written in INSERT DELAY queues
mysql	status	Open_files	long	The number of files that are open
mysql	status	Open_streams	long	The number of streams that are open (used mainly for logging)
mysql	status	Open_table_definitions	long	
mysql	status	Open_tables	long	The number of tables that are open
mysql	status	Opened_files	long_counter	
mysql	status	Opened_table_definitions	long_counter	

Namespace	Type	Attribute	Type	Description
mysql	status	Opened_tables	long_counter	The number of tables that have been opened
mysql	status	Prepared_stmt_count	long	The current number of prepared statements
mysql	status	Qcache_free_blocks	long	The number of free memory blocks in the query cache
mysql	status	Qcache_free_memory	long	The amount of free memory for the query cache
mysql	status	Qcache_hits	long_counter	The number of query cache hits
mysql	status	Qcache_inserts	long_counter	The number of query cache inserts
mysql	status	Qcache_lowmem_prunes	long_counter	Performance.Query Cache
mysql	status	Qcache_not_cached	long_counter	Performance.Query Cache
mysql	status	Qcache_queries_in_cache	long	The number of queries registered in the query cache
mysql	status	Qcache_total_blocks	long	The total number of blocks in the query cache
mysql	status	Questions	long_counter	The number of statements that clients have sent to the server
mysql	status	Rpl_status	long	The status of fail-safe replication (not yet implemented)
mysql	status	Select_full_join	long_counter	Performance.Selects
mysql	status	Select_full_range_join	long_counter	The number of joins that used a range search on a reference table
mysql	status	Select_range	long_counter	The number of joins that used ranges on the first table
mysql	status	Select_range_check	long_counter	The number of joins without keys that check for key usage after each row
mysql	status	Select_scan	long_counter	The number of joins that did a full scan of the first table
mysql	status	Slave_open_temp_tables	long	Networking.Replication
mysql	status	Slave_retried_transactions	long_counter	Networking.Replication
mysql	status	Slave_running	string	This is ON if this server is a slave that 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

Namespace	Type	Attribute	Type	Description
mysql	status	Ssl_ctx_verify_depth	long	The SSL context verification depth
mysql	status	Ssl_ctx_verify_mode	long	The SSL certificate verification mode used by the server
mysql	status	Ssl_default_timeout	long	The SSL default timeout
mysql	status	Ssl_finished_accepts	long_counter	The number of successful SSL connections to the server
mysql	status	Ssl_finished_connects	long_counter	The number of successful SSL slave connections to the server
mysql	status	Ssl_session_cache_hits	long_counter	The number of SSL session cache hits
mysql	status	Ssl_session_cache_misses	long_counter	The number of SSL session cache misses
mysql	status	Ssl_session_cache_mode	string	The SSL session cache mode
mysql	status	Ssl_session_cache_overflows	long_counter	The number of SSL cache overflows
mysql	status	Ssl_session_cache_size	long	The size of the SSL session cache
mysql	status	Ssl_session_cache_timeouts	long_counter	The number of session cache timeouts
mysql	status	Ssl_sessions_reused	long	The number of SSL sessions from the cache that were reused
mysql	status	Ssl_used_session_cache_entries	long	The number of SSL session cache entries used
mysql	status	Ssl_verify_depth	long	Depth of verification for replication SSL connections
mysql	status	Ssl_verify_mode	long	Mode of verification 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	strong_privileges	user	string	
mysql	table	num_rows	long	MySQL Network.Table Data
mysql	table	table_auto_increment	long	The next AUTO_INCREMENT value
mysql	table	table_avg_row_length	long	The average row length
mysql	table	table_collation	string	The table's character set and collation
mysql	table	table_comment	string	The table comment
mysql	table	table_create_time	string	When the data file was created
mysql	table	table_data_free	long	The number of allocated but unused bytes
mysql	table	table_data_length	long	The length of the data file
mysql	table	table_engine	string	The storage engine used by a table
mysql	table	table_index_length	long	The length of the index file
mysql	table	table_max_data_length	long	The maximum length of the data file

Namespace	Type	Attribute	Type	Description
mysql	table	table_name	string	The name of a table
mysql	table	table_row_format	string	The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact)
mysql	table	table_version	long	The version number of the table's .frm file
mysql	tablestatus	Auto_increment	long	The next AUTO_INCREMENT value
mysql	tablestatus	Avg_row_length	long	The average row length
mysql	tablestatus	Check_time	string	When the table was last checked
mysql	tablestatus	Checksum	string	The live checksum value (if any)
mysql	tablestatus	Collation	string	The table's character set
mysql	tablestatus	Comment	string	The comment used when creating the table
mysql	tablestatus	Create_options	string	Extra options used with CREATE TABLE
mysql	tablestatus	Create_time	string	When the table was created
mysql	tablestatus	Data_free	long	The number of allocated but unused bytes
mysql	tablestatus	Data_length	long	The length of the data file
mysql	tablestatus	Engine	string	The storage engine for the table
mysql	tablestatus	Index_length	long	The length of the index file
mysql	tablestatus	Max_data_length	long	The maximum length of the data file
mysql	tablestatus	Name	string	The table name
mysql	tablestatus	Row_format	string	The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact).
mysql	tablestatus	Rows	long	The number of rows in the table
mysql	tablestatus	Update_time	string	When the data file was last updated
mysql	tablestatus	Version	long	The version number of the table's .frm file.
mysql	test_database	Database (test)	string	
mysql	transactions_in_serial_log	transactions_in_serial_log	long	
mysql	trigger_with_select_star	trigger_defn	string	
mysql	user_on_missing_db	db_name	string	
mysql	user_on_missing_db	user	string	
mysql	variables	auto_increment_increment	long	Auto-increment columns are incremented by this value
mysql	variables	auto_increment_offset	long	Offset added to Auto-increment columns
mysql	variables	autocommit	string	
mysql	variables	automatic_sp_privileges	string	Creating and dropping stored procedures alters ACLs
mysql	variables	back_log	long	The number of outstanding connection requests MySQL can have
mysql	variables	basedir	string	General.Directories / Files
mysql	variables	big_tables	string	
mysql	variables	binlog_cache_size	long	Memory.Caches
mysql	variables	binlog_format	string	
mysql	variables	bulk_insert_buffer_size	long	Size of tree cache used in bulk insert optimization
mysql	variables	character_set_client	string	Current client character set
mysql	variables	character_set_connection	string	Current connection character set

Namespace	Type	Attribute	Type	Description
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	
mysql	variables	event_scheduler	string	
mysql	variables	expire_logs_days	long	General.Miscellaneous
mysql	variables	flush	string	Flush tables to disk between SQL commands
mysql	variables	flush_time	long	A dedicated thread is created to flush all tables at the given interval
mysql	variables	foreign_key_checks	string	
mysql	variables	ft_boolean_syntax	string	List of operators for MATCH ... AGAINST (... IN BOOLEAN MODE)
mysql	variables	ft_max_word_len	long	SQL.Full Text Search
mysql	variables	ft_min_word_len	long	SQL.Full Text Search
mysql	variables	ft_query_expansion_limit	long	Number of best matches to use for query expansion
mysql	variables	ft_stopword_file	string	Use stopwords from this file instead of built-in list
mysql	variables	general_log	string	
mysql	variables	general_log_file	string	
mysql	variables	group_concat_max_len	long	The maximum length of the result of function group_concat
mysql	variables	have_archive	string	Whether mysqld supports archive tables
mysql	variables	have_bdb	string	Is Berkeley DB supported
mysql	variables	have_blackhole_engine	string	Whether mysqld supports BLACK-

Namespace	Type	Attribute	Type	Description
				HOLE tables
mysql	variables	have_community_features	string	
mysql	variables	have_compress	string	Availability of the zlib compression library
mysql	variables	have_crypt	string	Availability of the crypt() system call
mysql	variables	have_csv	string	Whether mysqld supports csv tables
mysql	variables	have_dynamic_loading	string	Whether mysqld supports dynamic loading of plugins
mysql	variables	have_example_engine	string	Whether mysqld supports EXAMPLE tables
mysql	variables	have_federated_engine	string	Whether mysqld supports FEDERATED tables
mysql	variables	have_geometry	string	Whether mysqld supports spatial data types
mysql	variables	have_innodb	string	Whether mysqld supports InnoDB tables
mysql	variables	have_isam	string	Whether mysqld supports isam tables
mysql	variables	have_merge_engine	string	Whether mysqld supports merge tables
mysql	variables	have_ndbcluster	string	Whether mysqld supports NDB Cluster tables
mysql	variables	have_openssl	string	Whether mysqld supports ssl connections
mysql	variables	have_partitioning	string	
mysql	variables	have_query_cache	string	Whether mysqld supports query cache
mysql	variables	have_raid	string	Whether mysqld supports the RAID option
mysql	variables	have_rtree_keys	string	General.Features
mysql	variables	have_ssl	string	Whether the server supports an SSL connection
mysql	variables	have_symlink	string	Is symbolic link support enabled
mysql	variables	hostname	string	The name of the server host
mysql	variables	identity	long	
mysql	variables	init_connect	string	Command(s) that are executed for each new connection
mysql	variables	init_file	string	Read SQL commands from this file at startup
mysql	variables	init_slave	string	Command(s) that are executed when a slave connects to a master
mysql	variables	innodb_adaptive_hash_index	string	
mysql	variables	innodb_additional_mem_pool_size	long	Memory.Buffers
mysql	variables	innodb_autoextend_increment	long	Data file autoextend increment in megabytes
mysql	variables	innodb_autoinc_lock_mode	long	
mysql	variables	innodb_buffer_pool_awesome_memory_mb	long	Memory.Buffers
mysql	variables	innodb_buffer_pool_size	long	Memory.Buffers
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

Namespace	Type	Attribute	Type	Description
mysql	variables	innodb_data_home_dir	string	The common part for InnoDB table spaces
mysql	variables	innodb_doublewrite	string	Enable InnoDB doublewrite buffer
mysql	variables	innodb_fast_shutdown	long	Speeds up the shutdown process of the InnoDB storage engine
mysql	variables	innodb_file_io_threads	long	Number of file I/O threads in InnoDB
mysql	variables	innodb_file_per_table	string	Stores each InnoDB table to an .ibd file in the database dir
mysql	variables	innodb_flush_log_at_trx_commit	long	Table Types.InnoDB
mysql	variables	innodb_flush_method	string	With which method to flush data
mysql	variables	innodb_force_recovery	long	Table Types.InnoDB
mysql	variables	innodb_lock_wait_timeout	long	Table Types.InnoDB
mysql	variables	innodb_locks_unsafe_for_binlog	string	Table Types.InnoDB
mysql	variables	innodb_log_arch_dir	string	Where full logs should be archived
mysql	variables	innodb_log_archive	string	Set to 1 if you want to have logs archived
mysql	variables	innodb_log_buffer_size	long	General.Logs
mysql	variables	innodb_log_file_size	long	Size of each log file in a log group
mysql	variables	innodb_log_files_in_group	long	General.Logs
mysql	variables	innodb_log_group_home_dir	string	Path to InnoDB log files
mysql	variables	innodb_max_dirty_pages_pct	long	Percentage of dirty pages allowed in bufferpool
mysql	variables	innodb_max_purge_lag	long	Desired maximum length of the purge queue (0 = no limit)
mysql	variables	innodb_mirrored_log_groups	long	Number of identical copies of log groups we keep for the database
mysql	variables	innodb_open_files	long	The maximum number of files that InnoDB keeps open at the same time
mysql	variables	innodb_rollback_on_timeout	string	Unknown.Unknown
mysql	variables	innodb_support_xa	string	Enable InnoDB support for the XA two-phase commit
mysql	variables	innodb_sync_spin_loops	long	Count of spin-loop rounds in InnoDB mutexes
mysql	variables	innodb_table_locks	string	Enable InnoDB locking in LOCK TABLES
mysql	variables	innodb_thread_concurrency	long	Table Types.InnoDB
mysql	variables	innodb_thread_sleep_delay	long	Table Types.InnoDB
mysql	variables	insert_id	long	
mysql	variables	interactive_timeout	long	Connections.Overview
mysql	variables	join_buffer_size	long	The size of the buffer that is used for full joins
mysql	variables	keep_files_on_create	string	
mysql	variables	key_buffer_size	long	The size of the buffer used for index blocks for MyISAM tables
mysql	variables	key_cache_age_threshold	long	Memory.Caches
mysql	variables	key_cache_block_size	long	The default size of key cache blocks
mysql	variables	key_cache_division_limit	long	The minimum percentage of warm blocks in key cache
mysql	variables	language	string	Client error messages in given language. May be given as a full path
mysql	variables	large_files_support	string	Whether large files are supported

Namespace	Type	Attribute	Type	Description
mysql	variables	large_page_size	long	General.Features
mysql	variables	large_pages	string	Enable support for large pages
mysql	variables	last_insert_id	long	
mysql	variables	lc_time_names	string	General.Miscellaneous
mysql	variables	license	string	Type of license for the server
mysql	variables	local_infile	string	Enable/disable LOAD DATA LOCAL INFILE (takes values 1 0)
mysql	variables	locked_in_memory	long	Whether mysqld is locked in memory with --memlock
mysql	variables	log	string	Log connections and queries to file
mysql	variables	log_bin	string	Enables binary log
mysql	variables	log_bin_trust_function_creators	string	General.Logs
mysql	variables	log_bin_trust_routine_creators	string	(deprecated) Use log-bin-trust-function-creators
mysql	variables	log_error	string	Error log file
mysql	variables	log_output	string	
mysql	variables	log_queries_not_using_indexes	string	General.Logs
mysql	variables	log_slave_updates	string	General.Logs
mysql	variables	log_slow_queries	string	General.Logs
mysql	variables	log_update	string	General.Logs
mysql	variables	log_warnings	long	Log some not critical warnings to the log file
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

Namespace	Type	Attribute	Type	Description
mysql	variables	max_tmp_tables	long	Maximum number of temporary tables a client can keep open at a time
mysql	variables	max_user_connections	long	Connections.Overview
mysql	variables	max_write_lock_count	long	After this many write locks, allow some read locks to run in between
mysql	variables	min_examined_row_limit	long	
mysql	variables	multi_range_count	long	General.Performance
mysql	variables	myisam_data_pointer_size	long	Default pointer size to be used for MyISAM tables
mysql	variables	myisam_max_extra_sort_file_size	long	Deprecated option
mysql	variables	myisam_max_sort_file_size	long	Table Types.MyISAM
mysql	variables	myisam_recover_options	string	The value of the --myisam-recover option
mysql	variables	myisam_repair_threads	long	Table Types.MyISAM
mysql	variables	myisam_sort_buffer_size	long	Memory.Buffers
mysql	variables	myisam_stats_method	string	Specifies how MyISAM index statistics collection code should treat NULLs
mysql	variables	myisam_use_mmap	string	
mysql	variables	named_pipe	string	Indicates whether the server supports connections over named pipes
mysql	variables	net_buffer_length	long	Buffer length for TCP/IP and socket communication
mysql	variables	net_read_timeout	long	Connections.Networking
mysql	variables	net_retry_count	long	Connections.Networking
mysql	variables	net_write_timeout	long	Connections.Networking
mysql	variables	new	string	Use very new, possibly 'unsafe' functions
mysql	variables	old	string	
mysql	variables	old_alter_table	string	
mysql	variables	old_passwords	string	Use old password encryption method (needed for 4.0 and older clients)
mysql	variables	open_files_limit	long	General.Directories / Files
mysql	variables	optimizer_prune_level	long	General.Features
mysql	variables	optimizer_search_depth	long	Maximum depth of search performed by the query optimizer
mysql	variables	pid_file	string	Pid file used by safe_mysql
mysql	variables	plugin_dir	string	
mysql	variables	port	long	Port number to use for connection
mysql	variables	preload_buffer_size	long	The size of the buffer that is allocated when preloading indexes
mysql	variables	prepared_stmt_count	long	The current number of prepared statements
mysql	variables	protocol_version	long	The version of the client/server protocol used by the MySQL server
mysql	variables	pseudo_thread_id	long	
mysql	variables	query_alloc_block_size	long	Allocation block size for query parsing and execution
mysql	variables	query_cache_limit	long	Don't cache results that are bigger than this
mysql	variables	query_cache_min_res_unit	long	Memory.Caches
mysql	variables	query_cache_size	long	The memory allocated to store results from old queries

Namespace	Type	Attribute	Type	Description
mysql	variables	query_cache_type	string	Query cache type
mysql	variables	query_cache_wlock_invalidate	string	Invalidate queries in query cache on LOCK for write
mysql	variables	query_prealloc_size	long	Persistent buffer for query parsing and execution
mysql	variables	rand_seed1	long	
mysql	variables	rand_seed2	long	
mysql	variables	range_alloc_block_size	long	Allocation block size for storing ranges during optimization
mysql	variables	read_buffer_size	long	Memory.Buffers
mysql	variables	read_only	string	Make all non-temporary tables read-only
mysql	variables	read_rnd_buffer_size	long	Memory.Buffers
mysql	variables	relay_log	string	
mysql	variables	relay_log_index	string	
mysql	variables	relay_log_info_file	string	
mysql	variables	relay_log_purge	string	Determines whether relay logs are purged
mysql	variables	relay_log_space_limit	long	Maximum space to use for all relay logs
mysql	variables	report_host	string	
mysql	variables	report_password	string	
mysql	variables	report_port	long	
mysql	variables	report_user	string	
mysql	variables	rpl_recovery_rank	long	Not used
mysql	variables	secure_auth	string	Disallow authentication for accounts that have old (pre-4.1) passwords
mysql	variables	secure_file_priv	string	Unknown.Unknown
mysql	variables	server_id	long	General.Overview
mysql	variables	shared_memory	string	Whether the server allows shared-memory connections
mysql	variables	shared_memory_base_name	string	The name of shared memory to use for shared-memory connections
mysql	variables	skip_external_locking	string	Skip system (external) locking
mysql	variables	skip_networking	string	Don't allow connection with TCP/IP
mysql	variables	skip_show_database	string	Don't allow 'SHOW DATABASE' commands
mysql	variables	slave_compressed_protocol	string	Use compression on master/slave protocol
mysql	variables	slave_exec_mode	string	
mysql	variables	slave_load_tmpdir	string	General.Replication
mysql	variables	slave_net_timeout	long	Connections.Networking
mysql	variables	slave_skip_errors	string	General.Replication
mysql	variables	slave_transaction_retries	long	General.Replication
mysql	variables	slow_launch_time	long	General.Performance
mysql	variables	slow_query_log	string	
mysql	variables	slow_query_log_file	string	
mysql	variables	socket	string	Socket file to use for connection
mysql	variables	sort_buffer_size	long	Each thread that needs to do a sort allocates a buffer of this size
mysql	variables	sql_auto_is_null	string	
mysql	variables	sql_big_selects	string	General.Features

Namespace	Type	Attribute	Type	Description
mysql	variables	sql_big_tables	string	
mysql	variables	sql_buffer_result	string	
mysql	variables	sql_log_bin	string	
mysql	variables	sql_log_off	string	
mysql	variables	sql_log_update	string	
mysql	variables	sql_low_priority_updates	string	
mysql	variables	sql_max_join_size	long	
mysql	variables	sql_mode	string	Set the SQL server mode
mysql	variables	sql_notes	string	If set to 1, warnings of Note level are recorded
mysql	variables	sql_quote_show_create	string	
mysql	variables	sql_safe_updates	string	
mysql	variables	sql_select_limit	long	
mysql	variables	sql_slave_skip_counter	long	
mysql	variables	sql_warnings	string	General.Miscellaneous
mysql	variables	ssl_ca	string	CA file in PEM format (check OpenSSL docs, implies --ssl)
mysql	variables	ssl_capath	string	CA directory (check OpenSSL docs, implies --ssl)
mysql	variables	ssl_cert	string	X509 certification in PEM format (implies --ssl)
mysql	variables	ssl_cipher	string	SSL cipher to use (implies --ssl)
mysql	variables	ssl_key	string	X509 key in PEM format (implies ssl)
mysql	variables	storage_engine	string	The default storage engine
mysql	variables	sync_binlog	long	Synchronously flush binary log to disk after every #th event
mysql	variables	sync_frm	string	Sync .frm to disk on create. Enabled by default
mysql	variables	sync_replication	long	Deprecated
mysql	variables	sync_replication_slave_id	long	Deprecated
mysql	variables	sync_replication_timeout	long	Deprecated
mysql	variables	system_time_zone	string	The server system time zone
mysql	variables	table_cache	long	Deprecated; use --table_open_cache instead
mysql	variables	table_definition_cache	long	
mysql	variables	table_lock_wait_timeout	long	General.Miscellaneous
mysql	variables	table_open_cache	long	
mysql	variables	table_type	string	A synonym for storage_engine
mysql	variables	thread_cache_size	long	How many threads we should keep in a cache for reuse
mysql	variables	thread_handling	string	
mysql	variables	thread_stack	long	The stack size for each thread
mysql	variables	time_format	string	The TIME format (for future)
mysql	variables	time_zone	string	The current time zone.
mysql	variables	timed_mutexes	string	General.Miscellaneous
mysql	variables	timestamp	long	
mysql	variables	tmp_table_size	long	Memory.Overview
mysql	variables	tmpdir	string	Path for temporary files
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

Namespace	Type	Attribute	Type	Description
				stored in binary log
mysql	variables	tx_isolation	string	The default transaction isolation level
mysql	variables	unique_checks	string	
mysql	variables	updatable_views_with_limit	string	SQL.Overview
mysql	variables	version	string	Output version information and exit
mysql	variables	version_comment	string	General.Overview
mysql	variables	version_compile_machine	string	The type of machine or architecture on which MySQL was built
mysql	variables	version_compile_os	string	The type of operating system on which MySQL was built
mysql	variables	wait_timeout	long	Connections.Overview
mysql	variables	warning_count	long	
mysql	waiting_db	waiting_db	string	
mysql	waiting_statement	waiting_statement	string	
mysql	waiting_thread	waiting_thread	long	
mysql	waiting_time	waiting_time	long	
mysql	waiting_user	waiting_user	string	
os	cpu	cpu_cache_size	long	The size of the CPU cache
os	cpu	cpu_idle	long_counter	The CPU idle time
os	cpu	cpu_mhz	long	The MHz rating of the CPU
os	cpu	cpu_name	string	The CPU name
os	cpu	cpu_sys	long_counter	CPU system / kernel time
os	cpu	cpu_user	long_counter	CPU user space time
os	cpu	cpu_vendor	string	The name of the CPU vendor
os	cpu	cpu_wait	long_counter	CPU IO wait time
os	disk	disk_bytes_read	long_counter	The number of bytes read from disk
os	disk	disk_bytes_written	long_counter	The number of bytes written to disk
os	disk	disk_queue	long	The size of the disk IO queue
os	disk	disk_reads	long_counter	The number of disk reads
os	disk	disk_rtime	long	
os	disk	disk_servicetime	double	
os	disk	disk_snaptime	long	
os	disk	disk_time	long	
os	disk	disk_writes	long_counter	The number of disk writes
os	disk	disk_wtime	long	
os	disk	fs_avail	long	The amount of available space for the mount point
os	disk	fs_files	long	The number of files on the mount point
os	disk	fs_free_files	long	The number of free files on the mount point
os	disk	fs_total	long	The total amount of space for the mount point
os	disk	fs_unused	long	The amount of unused space for the mount point
os	disk	fs_used	long	The amount of used space for the mount point
os	fs	fs_avail	long	
os	fs	fs_dev_name	string	
os	fs	fs_dir_name	string	
os	fs	fs_files	long	

Namespace	Type	Attribute	Type	Description
os	fs	fs_flags	long	
os	fs	fs_free_files	long	
os	fs	fs_options	string	
os	fs	fs_sys_type_name	string	
os	fs	fs_total	long	
os	fs	fs_type_name	string	
os	fs	fs_unused	long	
os	fs	fs_used	long	
os	Host	host_id	string	The host unique identifier
os	Host	name	string	
os	loadavg	0	double	
os	loadavg	1	double	
os	loadavg	2	double	
os	mem	ram_total	long	The total amount of available RAM
os	mem	ram_unused	long	The total amount of unused RAM
os	mem	swap_page_in	long	
os	mem	swap_page_out	long	
os	mem	swap_total	long	The total amount of available swap memory
os	mem	swap_unused	long	The total amount of unused swap memory
os	net	address	string	The address of the network interface
os	net	broadcast	string	The network interface broadcast address
os	net	description	string	The network interface description
os	net	destination	string	The network interface destination address
os	net	flags	long	The network interface flags
os	net	hwaddr	string	The network interface MAC address
os	net	metric	long	Operating System.Net
os	net	mtu	long	The Maximum Transmission Unit
os	net	name	string	The network interface name (eth0, eth1 etc.)
os	net	netmask	string	The network interface subnet mask
os	net	rx_bytes	long_counter	The number of bytes received
os	net	rx_dropped	long_counter	The number of received packets dropped
os	net	rx_errors	long_counter	The number of received packet errors
os	net	rx_frame	long_counter	The number received packet frame errors
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

Namespace	Type	Attribute	Type	Description
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