MySQL Enterprise Monitor 3.0.27 Manual
Abstract

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

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

This manual documents the MySQL Enterprise Monitor version 3.0.27.

For notes detailing the changes in each release, see the MySQL Enterprise Monitor 3.0 Release Notes.

For legal information, see the Legal Notice.

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

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

**Licensing information.** This product may include third-party software, used under license. See this document for licensing information, including licensing information relating to third-party software that may be included in this release of MySQL Enterprise Monitor.

Document generated on: 2017-05-08 (revision: 6896)
# Table of Contents

Preface and Legal Notices .................................................................................................................. xi
I Getting Started with MySQL Enterprise Monitor .................................................................................. 1
   1 MySQL Enterprise Monitor Introduction and Architecture ................................................................. 5
       1.1 MySQL Enterprise Monitor Component Overview .............................................................................. 5
       1.2 MySQL Enterprise Monitor Agent .................................................................................................. 6
       1.3 MySQL Enterprise Service Manager ............................................................................................... 7
       1.4 MySQL Enterprise Monitor Proxy and Aggregator .......................................................................... 9
   2 What's new in MySQL Enterprise Monitor 3.0? .................................................................................. 11
II Installing MySQL Enterprise Monitor ................................................................................................. 15
   3 Installation Prerequisites ..................................................................................................................... 19
       3.1 Installer Files .................................................................................................................................. 19
       3.2 Prerequisites .................................................................................................................................. 20
           3.2.1 System Requirements ............................................................................................................... 20
           3.2.2 Supported Platforms ............................................................................................................... 21
           3.2.3 MySQL Enterprise Monitor Repository ................................................................................... 22
       3.3 Credentials Needed for Installation ................................................................................................. 23
           3.3.1 Existing Users .......................................................................................................................... 23
           3.3.2 Users Created During Installation ........................................................................................... 23
           3.3.3 Users Created on First Log-in .................................................................................................. 24
   4 Service Manager Installation ................................................................................................................ 25
       4.1 MySQL Enterprise Monitor Installation Types .................................................................................. 25
       4.2 MySQL Enterprise Service Manager Graphical Installation Wizard ................................................. 26
       4.3 Text-Based Installation .................................................................................................................... 29
       4.4 Starting/Stopping the MySQL Enterprise Monitor Services ........................................................... 29
       4.5 MySQL Enterprise Service Manager Configuration Settings ...................................................... 30
           4.5.1 Initial Monitor UI Log-In ........................................................................................................... 31
           4.5.2 Setting the Timezone and Locale .............................................................................................. 32
   5 Monitor Agent Installation .................................................................................................................... 33
       5.1 General Agent Related Notes ......................................................................................................... 33
       5.2 Creating MySQL User Accounts for the Monitor Agent ................................................................. 34
       5.3 Java Considerations on Linux .......................................................................................................... 37
       5.4 Installing the Monitor Agent on Unix and Linux ............................................................................. 37
       5.5 Installing the Agent on Microsoft Windows ...................................................................................... 42
       5.6 Installing the Agent on Mac OS X .................................................................................................... 45
       5.7 Starting/Stopping the MySQL Enterprise Monitor Agent ............................................................... 52
           5.7.1 Starting/Stopping the Agent on Windows .................................................................................... 53
           5.7.2 Starting/Stopping the Agent on Mac OS X ............................................................................... 54
           5.7.3 Starting/Stopping the Agent on Unix ......................................................................................... 55
           5.7.4 sql_mode .................................................................................................................................. 56
       5.8 Monitoring Multiple MySQL Servers .............................................................................................. 56
       5.9 Configuring an Agent to Monitor a Remote MySQL Server ............................................................. 56
       5.10 Monitoring Outside the Firewall with an SSH Tunnel ..................................................................... 57
       5.11 HTTP Connection Timeout ........................................................................................................... 58
       5.12 Troubleshooting the Agent ............................................................................................................. 58
       5.13 Agent Backlog .................................................................................................................................. 59
   6 Post-installation Considerations ........................................................................................................ 61
       6.1 General Considerations .................................................................................................................... 61
       6.2 Installing SSL Certificates .............................................................................................................. 62
       6.3 Changing an SSH Host Key .............................................................................................................. 65
   7 Upgrading, Re-Installing or Changing Your Installation ....................................................................... 67
       7.1 General considerations when upgrading MySQL Enterprise Monitor ........................................... 67
       7.2 Restoring from Backup .................................................................................................................... 69
       7.3 Guide for Upgrading to MySQL Enterprise Monitor 3.0 ................................................................. 69
       7.4 Upgrading an Existing 3.0.x Installation .......................................................................................... 72
   8 Unattended Installation Reference ..................................................................................................... 75
8.1 Unattended Installation ................................................................. 75
  8.1.1 Performing an Unattended Installation ................................. 75
  8.1.2 MySQL Enterprise Service Manager Options ........................... 76
  8.1.3 MySQL Enterprise Monitor Agent Options .............................. 82
9 Performance Tuning MySQL Enterprise Monitor .................................. 91
  9.1 Tuning Memory ................................................................. 91
  9.2 Tuning CPU ................................................................. 92
  9.3 Tuning Apache Tomcat Threads .............................................. 93
  9.4 Tuning Agent Memory Requirements ....................................... 94
10 Uninstalling the MySQL Enterprise Monitor ..................................... 95
  10.1 Removing the MySQL Enterprise Monitor: Windows ....................... 95
  10.2 Removing the MySQL Enterprise Monitor: Unix ......................... 96
  10.3 Removing the MySQL Enterprise Monitor Mac OS X ..................... 98
  10.4 Unattended Uninstallations ................................................. 99
11 Proxy and Aggregator Installation ................................................. 101
  11.1 Proxy Aggregator Architecture ............................................. 101
  11.2 Prerequisites ............................................................. 102
  11.3 Installing the Proxy and Aggregator ....................................... 103
  11.4 Graphical Installation Wizard ............................................. 103
  11.5 Text-Based Installation ................................................... 105
  11.6 Unattended Installation ................................................... 106
  11.7 Starting and Stopping the Proxy and Aggregator ......................... 108
  11.8 Configuration Options .................................................... 109
12 Configuring Connectors .................................................................. 113
  12.1 Using the MySQL Enterprise Plugin for Connector/PHP ................. 113
  12.2 Using the MySQL Enterprise Plugin for Connector/J .................. 117
  12.3 Using the MySQL Enterprise Plugin for Connector/Net ............... 121
13 The User Interface ......................................................................... 129
  13.1 General ............................................................................ 129
    13.1.1 Status Summary .......................................................... 129
    13.1.2 Asset Selector ............................................................ 130
  13.2 The Query Analyzer ........................................................... 132
  13.3 What's New ..................................................................... 132
14 The Dashboards ............................................................................ 135
  14.1 Introduction ......................................................................... 135
  14.2 Overview ........................................................................... 135
  14.3 Replication ......................................................................... 138
  14.4 MySQL Instances Dashboard ................................................ 140
    14.4.1 MySQL Instance Dashboard UI ....................................... 140
    14.4.2 MySQL Instance Details ............................................... 142
    14.4.3 Adding Instances ....................................................... 144
    14.4.4 Filtering MySQL Instances .......................................... 146
15 Settings ..................................................................................... 149
  15.1 Global Settings ..................................................................... 149
  15.2 Manage Users ..................................................................... 155
  15.3 Logs .................................................................................. 156
  15.4 User Preferences .................................................................. 158
  15.5 Diagnostics Report ................................................................ 159
16 Reports and Graphs ..................................................................... 161
  16.1 All Timeseries Graphs .......................................................... 161
  16.2 InnoDB Buffer Pool Usage .................................................... 162
17 Events ........................................................................................ 165
  17.1 Closing an Event .................................................................. 167
  17.2 Notification of Events .......................................................... 168
18 Event Handling ............................................................................ 171
  18.1 Event Handling Page ............................................................ 171
    18.1.1 Event Handlers List ....................................................... 171
List of Figures

1.1 MySQL Enterprise Monitor Architecture ................................................................. 5
1.2 MySQL Enterprise Monitor Agentless Architecture .................................................... 6
4.1 Initial setup for the MySQL Enterprise Monitor User Interface .................................. 31
5.1 Installing Agent on Windows: Host options ............................................................... 43
5.2 MySQL Enterprise Monitor: Installing Agent on Windows: Agent Authentication ........ 44
5.3 MySQL Enterprise Monitor: Installing Agent on Windows: Monitored Database Information 44
5.4 MySQL Enterprise Monitor: Installing Agent on Windows: Additional Agent Users ....... 45
5.5 MySQL Enterprise Monitor: Agent Monitoring Options ............................................. 47
5.6 MySQL Enterprise Monitor: Install as a service ....................................................... 48
5.7 MySQL Enterprise Monitor: Installing Agent on Mac OS X: MySQL Enterprise Service 49
Manager Options .............................................................................................................. 49
5.8 MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information (TCIP) .................................................................................................................. 50
5.9 MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information (Socket) .................................................................................................................. 51
5.10 MySQL Enterprise Monitor: Installing Agent on OS X: Additional Agent Users .......... 52
11.1 MySQL Enterprise Monitor Proxy and Aggregator Architecture ............................... 102
12.1 Plugin for PHP and Aggregator Architecture ............................................................ 113
12.2 Connector Plugin Architecture .................................................................................. 117
12.3 Connector Plugin Architecture .................................................................................. 122
13.1 A Status Summary Example ....................................................................................... 129
13.2 The What's New Page ................................................................................................. 132
13.3 What's New Configuration ......................................................................................... 134
14.1 Group Overview Configuration .................................................................................. 135
14.2 MySQL Instances: Group Context Menu .................................................................... 143
14.3 Add Instance Connection Settings ............................................................................ 144
14.4 Add Instance Advanced Settings .............................................................................. 145
14.5 MySQL Instance Filter ............................................................................................... 147
15.1 MySQL Enterprise Monitor User Interface Settings: Server Locale .......................... 149
15.2 MySQL Enterprise Monitor User Interface Settings: Server Hostname ...................... 150
15.3 MySQL Enterprise Monitor User Interface Settings: Data Purge ............................... 150
15.4 MySQL Enterprise Monitor User Interface Settings: My Oracle Support Credentials 151
15.5 MySQL Enterprise Monitor User Interface Settings: HTTP Proxy Settings ............... 152
15.6 MySQL Enterprise Monitor User Interface Settings: Customize MySQL server name 154
15.7 MySQL Enterprise Monitor User Interface: Manage Users ...................................... 155
15.8 MySQL Enterprise Monitor User Interface: Logs ...................................................... 157
15.9 MySQL Enterprise Monitor User Interface: User Preferences .................................. 158
17.1 MySQL Enterprise Monitor User Interface: Events Screen with Search enabled ....... 166
18.1 Event Handlers section ............................................................................................... 171
18.2 Email Notification Groups section ............................................................................. 172
18.3 Create Group Dialog ............................................................................................... 173
18.4 Email Settings section ............................................................................................. 174
18.5 SNMP Settings section ............................................................................................. 174
18.6 Create Event Handler Dialog ................................................................................... 176
19.1 Advisors Page ........................................................................................................... 179
19.2 Advisor Menu Control ............................................................................................. 181
19.3 Advisor Popup Menu ............................................................................................... 181
19.4 Advisor Popup Menu ............................................................................................... 182
19.5 Agent Health - General ............................................................................................ 184
19.6 Threshold Definitions Example ................................................................................ 185
21.1 Agent Health - General ............................................................................................ 221
21.2 Agent Health - General ............................................................................................ 222
21.3 Agent Health - Backlog ............................................................................................. 223
21.4 CPU Usage .............................................................................................................. 226
21.5 CPU Outliers ........................................................................................................... 226
List of Tables

3.1 Disk space Required ................................................................. 20
4.1 Installation Parameters ............................................................ 26
6.1 Default Root Path ($INSTALL_ROOT) ........................................... 62
6.2 SSL Configuration Options For The Agent's bootstrap.properties ........ 64
8.1 MySQL Enterprise Service Manager Installer Options .................... 76
8.2 MySQL Enterprise Monitor Agent Installer Options ....................... 82
9.1 Apache Tomcat configuration file location (default) ..................... 91
9.2 Installation Parameters ............................................................ 91
9.3 MEM repository configuration file location (default) ..................... 92
9.4 MEM repository configuration tool location (default) .................... 93
9.5 MEM repository configuration tool location (default) .................... 93
10.1 MySQL Enterprise Monitor Uninstaller Options .......................... 99
11.1 MySQL Enterprise Monitor Proxy and Aggregator Installer Options .... 106
11.2 Proxy and Aggregator Help Options ........................................... 110
11.3 Application Options ............................................................... 110
11.4 aggr-module Options ............................................................... 111
11.5 proxy-module Options ............................................................. 111
12.1 Connector/PHP Properties ......................................................... 116
12.2 MySQL Plugin for Connector/J Properties ................................... 118
12.3 MySQL Plugin for Connector/J SSL Properties ............................ 119
14.1 Group Overview Configuration .................................................. 135
14.2 Bad Connection List ............................................................... 141
14.3 Unreachable Agents List .......................................................... 141
14.4 Unmonitored MySQL Instances List ......................................... 142
14.5 MySQL Instance Details Columns .............................................. 142
14.6 Connection Settings Tab .......................................................... 144
14.7 Advanced Settings ................................................................. 146
14.8 MySQL Instance Filter ............................................................. 147
18.1 Event Handler List Controls ...................................................... 171
18.2 Email Notification Groups Controls .......................................... 172
18.3 Email Settings Controls ........................................................... 174
18.4 SNMP Settings Controls ........................................................... 175
18.5 Create Event Handler Controls ................................................ 176
19.1 Advisor Page Controls ............................................................. 179
19.2 Advisor Information Listing ..................................................... 181
19.3 Advisor Edit Menu Controls ..................................................... 181
19.4 Advisor Filter Controls ............................................................ 182
21.1 MySQL Process Discovery Controls .......................................... 224
22.1 MySQL Enterprise Monitor: Wiki Formatting .............................. 237
22.2 Custom Data Collection Class Elements .................................... 240
22.3 Attribute Elements ................................................................. 242
23.1 QRTI value definitions ............................................................. 255
D.1 MySQL Enterprise Monitor: Log File Locations ........................... 281
D.2 MySQL Enterprise Monitor: MIB File Locations ......................... 281
D.3 MySQL Enterprise Monitor: Default path of the config.properties File .................................................................................. 282
D.4 Optional config.properties values .............................................. 282
D.5 MEM Repository Configuration Tool Location (default) ................ 286
D.6 MEM Agent Configuration Tool Location (default) ....................... 287
E.1 MySQL Enterprise Dashboard: GUI Error Codes and Messages ....... 289
E.2 MySQL Enterprise Monitor: Server Codes and Messages ............. 294
G.1 MySQL Monitor Data Items .......................................................... 301
Preface and Legal Notices

This manual documents the MySQL Enterprise Monitor version 3.0.27.

**Licensing information.** This product may include third-party software, used under license. See this document for licensing information, including licensing information relating to third-party software that may be included in this release of MySQL Enterprise Monitor.

**Legal Notices**

Copyright © 2005, 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

**U.S. GOVERNMENT END USERS:** Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

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 Oracle 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 Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.
Part I Getting Started with MySQL Enterprise Monitor
Table of Contents

1 MySQL Enterprise Monitor Introduction and Architecture ................................................................. 5
   1.1 MySQL Enterprise Monitor Component Overview ................................................................. 5
   1.2 MySQL Enterprise Monitor Agent ......................................................................................... 6
   1.3 MySQL Enterprise Service Manager ...................................................................................... 7
   1.4 MySQL Enterprise Monitor Proxy and Aggregator ................................................................. 9
2 What's new in MySQL Enterprise Monitor 3.0? ............................................................................. 11
Chapter 1 MySQL Enterprise Monitor Introduction and Architecture

Table of Contents

1.1 MySQL Enterprise Monitor Component Overview .................................................. 5
1.2 MySQL Enterprise Monitor Agent ........................................................................ 6
1.3 MySQL Enterprise Service Manager ................................................................. 7
1.4 MySQL Enterprise Monitor Proxy and Aggregator ............................................ 9

Important

This document is updated frequently. The most up-to-date version of this document is available at this location: MySQL Enterprise Products Documentation.

Note

MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

MySQL Enterprise Monitor is a companion product to MySQL Server that enables monitoring of MySQL instances and their hosts, notification of potential issues and problems, and advice on how to correct issues. MySQL Enterprise Monitor can monitor all types of installation, from a single MySQL instance to large farms of database servers. MySQL Enterprise Monitor is a web-based application, enabling you to monitor MySQL instances on your network or on a cloud service.

This chapter describes the components of a MySQL Enterprise Monitor installation and provides a high-level overview of MySQL Enterprise Monitor architecture.

1.1 MySQL Enterprise Monitor Component Overview

The architecture of a typical MySQL Enterprise Monitor installation is shown in the following figure:

Figure 1.1 MySQL Enterprise Monitor Architecture
MySQL Enterprise Monitor has the following components:

- MySQL Enterprise Monitor Agent monitors the MySQL instances and hosts, and collects data according to a defined schedule. The collection data is sent to the MySQL Enterprise Service Manager for analysis and presentation.

- MySQL Enterprise Service Manager analyzes, stores and presents the data collected by the agent.

- MySQL Enterprise Monitor Proxy and Aggregator intercepts queries as they are transmitted from client applications to the monitored MySQL instance and transmits them to the MySQL Enterprise Service Manager for analysis by the Query Analyzer.

It is also possible to perform what is called an Agent-less installation, where the Agent is not installed on the host machines, and all monitoring is done by the MySQL Enterprise Service Manager’s built-in Agent.

**Figure 1.2 MySQL Enterprise Monitor Agentless Architecture**

![MySQL Enterprise Monitor Agentless Architecture Diagram]

### 1.2 MySQL Enterprise Monitor Agent

The Agent collects data from the monitored instance and host, and transmits that data to the MySQL Enterprise Service Manager. The Agent can be installed on the same host as the MySQL instance or on a different host.

- Provides the direct monitoring of the MySQL server, including checking the server accessibility, configuration, obtaining the server ID, and setting up the environment to enable collecting more detailed information. In addition to the information accessible by accessing variable and configuration information within the server, other configuration parameters, such as the replication topology, are also collected from the server.

- Collects the operating system specific information, including RAM, disk storage and other data.

**Note**

The Agent can collect host data for the server on which it is installed, only. It cannot collect such data for a remotely monitored host.

- Collects the data from the server, including obtaining the values and configuration of the MySQL server, status variables and other information.
MySQL Enterprise Service Manager

• Communicates with the MySQL Enterprise Service Manager. Data is collected at scheduled intervals according to the schedule defined on the Advisors. This information is then sent to the MySQL Enterprise Service Manager.

• For MySQL 5.6.14 and greater, the Agent also collects digested query data from the Performance Schema and populates the Query Analyzer.

Important

If an Agent monitors a MySQL instance remotely, it cannot monitor the host and can only collect data from the monitored MySQL instance.

The Agent runs as a service. The data collected by the Agent is defined by enabling, or disabling, MySQL Enterprise Monitor Advisors.

1.3 MySQL Enterprise Service Manager

MySQL Enterprise Service Manager is the central hub of the MySQL Enterprise Monitor installation and is responsible for the following:

• Receiving and storing information from the Agents.

• Configuring the types of information collected by the Agents.

• Analyzing the collected data using the Advisors.

• Generating alerts and sending mail or SNMP notifications based on the Advisor configuration.

• Displaying the collected data, events and notifications.

• Graphing and reporting on the collected data.

• Analyzing the SQL queries performed on the monitored instance, in real-time, using the Query Analyzer.

MySQL Enterprise Service Manager is a web application which runs on the Apache Tomcat server.

MySQL Enterprise Service Manager also contains its own Agent which, in a default installation, is used to monitor the repository and host. It can also be used to monitor other, remote MySQL instances. This Agent is installed automatically, as part of the MySQL Enterprise Service Manager installation.

MySQL Enterprise Service Manager Repository

The repository is a MySQL instance which stores all data collected by the Agent. The majority of the data collected by the Agent is analyzed on-the-fly by the Advisors, then stored in the repository. The graphs and reports utilise the stored data to present information in the MySQL Enterprise Monitor User Interface.

MySQL Enterprise Monitor installer installs and configures the MySQL repository. It is also possible to use an existing MySQL instance for this purpose.

MySQL Enterprise Monitor User Interface

The MySQL Enterprise Monitor User Interface is a web-based interface to the MySQL Enterprise Service Manager. MySQL Enterprise Monitor User Interface provides a quick overview of the current status of your hosts and MySQL instances, and enables you to drill down into the current status, events, and historical information submitted by each MySQL Enterprise Monitor Agent.

The main features of the MySQL Enterprise Monitor User Interface include:
MySQL Enterprise Advisors

- A simple **Overview** dashboard that gives an overview of the current health and status of all assets, a list of top critical and emergency events that should be handled, and graphs that relay database statistical information.

- The **Configuration** page lets you customize the Advisors and Event Handling for your system. For example, this includes setting thresholds for Advisors, and email addresses to send alerts.

- The **Query Analyzer** page helps you identify problematic queries.

- The **Replication** dashboard monitors the structure and health of your replication environment.

- The **Events** page lists all monitored events, which can be sorted and searched.

- The **MySQL Instances** dashboard lists all monitored MySQL instances, which can be analyzed, configured, and grouped.

- The **Graphs & Reports** section includes graphs with compiled data for your system that are updated according to the chosen assets. This includes the **All Timeseries Graphs** and **InnoDB Buffer Pool Usage** graph pages.

- The **What's New?** tab gives a live connection to the My Oracle Support site, with news about the latest releases, critical fixes and patches, current service requests, and suggestions for completing your installation.

### MySQL Enterprise Advisors

Advisors filter and evaluate the information broadcast by the Monitoring Agents and present it to the Events page when the defined thresholds are breached. They also present advice on what caused the breach and how to correct it. There are more than 200 Advisors, all of which are enabled by default. Thresholds are the predefined limits for Advisors. If the monitored data breaches the defined threshold, an event is generated and displayed on the Events page. Advisor thresholds use a variety of different value types, depending on the monitored value. Some use percentages, such as percentage of maximum number of connections. Others use timed durations, such as the average statement execution time. It is also possible to check if specific configuration elements are present or correct.

The following types of Advisor are provided:

- **Administration**: Checks the MySQL instance installation and configuration.

- **Agent**: Checks the status of each MySQL Enterprise Monitor Agent.

- **Availability**: Checks the availability of the MySQL process and the connection load.

- **Backup**: Checks whether backup jobs succeed or fail, required resources, and information about MySQL Enterprise Backup specific tasks.

- **Cluster**: Checks the status of the monitored MySQL Cluster.

- **Graphing**: Data for graphs.

- **Memory Usage**: Indicate how efficiently you are using various memory caches, such as the InnoDB buffer pool, MyISAM key cache, query cache, table cache, and thread cache.

- **Monitoring and Support Services**: Advisors related to the MySQL Enterprise Monitoring services itself.

- **Operating System**: Checks the Host Operating System performance.

- **Performance**: Identifies potential performance bottlenecks, and suggests optimizations.

- **Query Analysis**: Advisors related to Queries and Query Analysis.

- **Replication**: Identifies replication bottlenecks, and suggests replication design improvements.
Events and Notifications

The MySQL Enterprise Service Manager alerts you of Threshold breaches in the following ways:

- **Events**: If an Advisor’s defined Threshold is breached, an Event is generated and displayed on the **Events** page. This is the default alert method.

- **Notifications**: MySQL Enterprise Service Manager can be configured to send alerts by e-mail, or SNMP traps. These methods must be configured and are not enabled by default.

Query Analyzer

The Query Analyzer enables you to monitor all SQL statements executed on the monitored MySQL databases. The query data can be provided in one of the following ways:

- **Performance Schema**: for monitored versions of MySQL 5.6.14 or higher, the Agent retrieves query information from the Performance Schema. For more information, see Section 23.1.1, “Using the MySQL Performance Schema”

- **MySQL Enterprise Monitor Aggregator**: aggregates raw query statistics taken directly from client connections, but analyzed out-of-band and transmitted to the MySQL Enterprise Service Manager saving memory and processing overhead for client statements. MySQL Enterprise Monitor Aggregator can provide data from the Connector/PHP or, when used with the MySQL Enterprise Monitor Proxy, directly from the client application.

For more information, see Chapter 11, **Proxy and Aggregator Installation**

- **MySQL Connectors**: combined with the corresponding MySQL Enterprise Plugin can provide tracing and statistical information directly to MySQL Enterprise Service Manager.

For more information, see Chapter 12, **Configuring Connectors**

**Important**

Currently, it is possible to use only one source for the Query Analyzer. That is, if you are using MySQL Enterprise Monitor Proxy and Aggregator, you must deactivate Performance Schema on the monitored instance. The same is true if you are using MySQL Connectors to aggregate query data for the Query Analyzer.

1.4 MySQL Enterprise Monitor Proxy and Aggregator

The MySQL Enterprise Monitor Aggregator collects and summarizes the raw query statistics sent from the client application. This data is sent to the MySQL Enterprise Service Manager where it populates the Query Analyzer.

The MySQL Enterprise Monitor Aggregator requires a framework, or chassis, to handle the communications between the client application and MySQL instance, and to enable the MySQL Enterprise Monitor Aggregator to communicate with the MySQL Enterprise Service Manager. The following frameworks are available:
• MySQL Enterprise Monitor Proxy: the Proxy functions as the communications chassis for the Aggregator and is responsible for intercepting the communications between the client application and the MySQL instance. This enables the Aggregator to collect the raw query data sent from the client application to the MySQL instance. The MySQL Enterprise Monitor Proxy and Aggregator installer can install and configure both Proxy and Aggregator, or a standalone Aggregator if one of the MySQL connectors is used as the communications chassis. The client application must be configured to communicate with the MySQL Enterprise Monitor Proxy.

• MySQL Connectors: the MySQL Connectors enable communication between the client application and the MySQL instance. If you intend to use a MySQL Connector as the communications framework for the MySQL Enterprise Monitor Aggregator, you must configure the Connector to communicate with the Aggregator. If you use a Connector with the Aggregator, you do not need to install the MySQL Enterprise Monitor Proxy.

Important

Currently, the Aggregator is only required by the MySQL Enterprise Plugin for Connector/PHP. The other connectors can be configured to communicate query data with MySQL Enterprise Service Manager and do not require MySQL Enterprise Monitor Aggregator.
Chapter 2 What's new in MySQL Enterprise Monitor 3.0?

This section highlights new functionality and is geared towards users of earlier versions of MySQL Enterprise Monitor.

If you are familiar with earlier versions of MySQL Enterprise Monitor, after one glance at the 3.0 UI you will immediately notice significant differences in functionality, organization and appearance. "Under the hood" changes are even more dramatic, with entirely different models and implementation for inventory, instruments, Query Analyzer, Advisor, Graphs and Event handling and notifications.

For information about upgrading MySQL Enterprise Monitor 2.3 to 3.0, see Section 7.3, “Guide for Upgrading to MySQL Enterprise Monitor 3.0”.

Easy to set up and configure

- Policies, Groups and auto-scheduling are all improved to make administration of scale-out easier and automatic:
  - Add a MySQL Instance to one or more Groups using the MySQL Instances dashboard or when installing the Agent (which can be scripted)
  - Members of a Group automatically inherit Advisor and Graph schedules and configurations (all of which are auto-scheduled on startup), see Section 22.1, “Customizing Groups”.
  - Event Handling and notifications are centrally managed, group-aware and de-coupled from Advisor scheduling. See Chapter 18, Event Handling.
  - Adaptive scheduling - Changes in your IT infrastructure are auto-detected. For example, collections, analysis and notifications automatically adapt when new servers or slaves are provisioned, a master becomes a slave, file systems are mounted, etc.
  - As a result, connections to new Hosts and MySQL Instances can be configured without need for manual administration.

- Zero Configuration Query Analyzer - Works “out of the box” with MySQL 5.6 Performance_Schema (supported by 5.6.14 or later).

- Host monitoring - A newly-installed Agent begins collecting local CPU, memory, file system, and other OS-related data whether or not the Agent has also been configured to monitor a MySQL Instance at install time.

- Auto-discovery of mysqld instances - Agent automatically detects local, unmonitored mysqld processes and reports them to the UI as unmonitored. You can ignore or add monitoring connections to them on the MySQL Instances Dashboard[2]. You can fully automate monitoring of newly discovered Instances by editing the connection parameters for the MySQL Process Discovery Advisor and enabling "Attempt Connection"[3].

- Centralized Agent configuration - Add, update, ignore monitoring connections from the UI without having to tunnel into the host remotely.

- Multi-instance monitoring - Conserve system resources on your monitored systems by installing a single Agent per host no matter how many MySQL Instances are running there.
Automatic Analysis & Visual Presentation

- Remote ("agent-less") monitoring - Installing an Agent on each Host is recommended, but only required for collecting OS-related data.

Automatic Analysis & Visual Presentation

- Trends, projections and forecasting - Graphs and Event handlers inform you in advance of impending file system capacity problems

- Database Availability - SLA reporting is made easy with a graphical presentation of database availability for the past day, week and month

- Highlights top problems - Ranks and presents Hosts and MySQL Instances with the most critical problems

- Query Response Time index Numeric rating offers immediate insight into query performance (graphically presented by query, by Group or overall)[1]. For more information, see Section 23.2, "Query Response Time index (QRTi)".

- Visual SQL/graph correlation - Drill into any region on a graph to view SQL executing during the selected time period[1].

- Expand any monitored Instance on the MySQL Instances Dashboard to browse its server configuration.

- Retrieve and view a live map of Innodb Buffer Pool usage.

- There are many additional UI improvements, including:
  - Dynamic page content updates using AJAX.
  - The Asset selector supports search (useful for large monitoring environments) and optionally show host and monitored assets.
  - Client-side graphing make graphs richer and more responsive, see Chapter 16, Reports and Graphs.
  - Most UI elements now include inline tooltips (hover to see their annotations)

- A new Status Summary is displayed on every page, updates dynamically, and shows current status counters for Hosts monitored, MySQL Instances monitored, MySQL Instances with invalid connection configurations, Unmonitored MySQL Instances, and Emergency Events. The counters are live links: click to navigate for details or to resolve the issues they're reporting.

New Advisor engine and UI

- Advisor configuration and execution have been greatly improved, with changes including:
  - A new "emergency" level to highlight outages, etc.
  - False positives from flapping or spikes are avoided using exponential moving averages and other statistical techniques.
  - Advisors can analyze data across an entire group; for example, the Replication Advisor can scan an entire topology to find common configuration errors like duplicate server UUIDs or a slave whose version is less than its master's.
  - Advanced rules can search for multiple problem conditions but avoid cascading errors by generating only a single Event and notification.
• The UI for Advisor scheduling and configuration centers around Groups and Group membership. Selecting a Group or individual MySQL Instance lets you view or customize its schedules, thresholds, etc. Any changes apply to both current and future Group members.

Footnotes

[1] Requires Query Analyzer, which now collects SQL performance data using the PERFORMANCE_SCHEMA from 5.6.14 or later. This release also supports Query Analyzer sources like the Connector plugins (C/Java, C/php, C/Net) or the MySQL Proxy. There are advantages and disadvantages to each approach, see Section 13.2, “The Query Analyzer” for a brief discussion of differences.


[3] To edit, click Configure, select Advisors, open the Monitoring and Support Services Advisor category and edit. Known limitation: This action is limited to a single set of credentials.
Part II Installing MySQL
Enterprise Monitor
# Table of Contents

3 Installation Prerequisites ........................................................................................................ 19
   3.1 Installer Files .................................................................................................................. 19
   3.2 Prerequisites .................................................................................................................... 20
      3.2.1 System Requirements ............................................................................................... 20
      3.2.2 Supported Platforms ............................................................................................... 21
      3.2.3 MySQL Enterprise Monitor Repository ................................................................... 22
   3.3 Credentials Needed for Installation .................................................................................. 23
      3.3.1 Existing Users ............................................................................................................ 23
      3.3.2 Users Created During Installation ........................................................................... 23
      3.3.3 Users Created on First Log-in ................................................................................ 24
4 Service Manager Installation .................................................................................................... 25
   4.1 MySQL Enterprise Monitor Installation Types .............................................................. 25
   4.2 MySQL Enterprise Service Manager Graphical Installation Wizard .............................. 26
   4.3 Text-Based Installation .................................................................................................... 29
   4.4 Starting/Stopping the MySQL Enterprise Monitor Services .......................................... 29
   4.5 MySQL Enterprise Service Manager Configuration Settings ....................................... 30
      4.5.1 Initial Monitor UI Log-In ......................................................................................... 31
      4.5.2 Setting the Timezone and Locale ............................................................................ 32
5 Monitor Agent Installation ...................................................................................................... 33
   5.1 General Agent Related Notes .......................................................................................... 33
   5.2 Creating MySQL User Accounts for the Monitor Agent ............................................... 34
   5.3 Java Considerations on Linux .......................................................................................... 37
   5.4 Installing the Monitor Agent on Unix and Linux ........................................................... 37
   5.5 Installing the Agent on Microsoft Windows ..................................................................... 42
   5.6 Installing the Agent on Mac OS X .................................................................................... 45
   5.7 Starting/Stopping the MySQL Enterprise Monitor Agent ............................................. 52
      5.7.1 Starting/Stopping the Agent on Windows ................................................................. 53
      5.7.2 Starting/Stopping the Agent on Mac OS X ............................................................... 54
      5.7.3 Starting/Stopping the Agent on Unix ....................................................................... 55
      5.7.4 sql_mode ................................................................................................................ 56
   5.8 Monitoring Multiple MySQL Servers .............................................................................. 56
   5.9 Configuring an Agent to Monitor a Remote MySQL Server .......................................... 56
   5.10 Monitoring Outside the Firewall with an SSH Tunnel ................................................... 57
   5.11 HTTP Connection Timeout ............................................................................................ 58
   5.12 Troubleshooting the Agent ............................................................................................ 58
   5.13 Agent Backlog ............................................................................................................... 59
6 Post-installation Considerations .............................................................................................. 61
   6.1 General Considerations .................................................................................................... 61
   6.2 Installing SSL Certificates .............................................................................................. 62
   6.3 Changing an SSH Host Key .............................................................................................. 65
7 Upgrading, Re-Installing or Changing Your Installation ......................................................... 67
   7.1 General considerations when upgrading MySQL Enterprise Monitor ............................ 67
   7.2 Restoring from Backup .................................................................................................... 69
   7.3 Guide for Upgrading to MySQL Enterprise Monitor 3.0 .............................................. 69
   7.4 Upgrading an Existing 3.0.x Installation ......................................................................... 72
8 Unattended Installation Reference .......................................................................................... 75
   8.1 Unattended Installation .................................................................................................... 75
      8.1.1 Performing an Unattended Installation .................................................................... 75
      8.1.2 MySQL Enterprise Service Manager Options ....................................................... 76
      8.1.3 MySQL Enterprise Monitor Agent Options ........................................................... 82
9 Performance Tuning MySQL Enterprise Monitor .................................................................... 91
   9.1 Tuning Memory ................................................................................................................ 91
   9.2 Tuning CPU .................................................................................................................... 92
   9.3 Tuning Apache Tomcat Threads ..................................................................................... 93
   9.4 Tuning Agent Memory Requirements .......................................................................... 94
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Uninstalling the MySQL Enterprise Monitor</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>10.1 Removing the MySQL Enterprise Monitor: Windows</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>10.2 Removing the MySQL Enterprise Monitor: Unix</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>10.3 Removing the MySQL Enterprise Monitor Mac OS X</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>10.4 Unattended Uninstallations</td>
<td>99</td>
</tr>
<tr>
<td>11</td>
<td>Proxy and Aggregator Installation</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>11.1 Proxy Aggregator Architecture</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>11.2 Prerequisites</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>11.3 Installing the Proxy and Aggregator</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>11.4 Graphical Installation Wizard</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>11.5 Text-Based Installation</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>11.6 Unattended Installation</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>11.7 Starting and Stopping the Proxy and Aggregator</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>11.8 Configuration Options</td>
<td>109</td>
</tr>
<tr>
<td>12</td>
<td>Configuring Connectors</td>
<td>113</td>
</tr>
<tr>
<td></td>
<td>12.1 Using the MySQL Enterprise Plugin for Connector/PHP</td>
<td>113</td>
</tr>
<tr>
<td></td>
<td>12.2 Using the MySQL Enterprise Plugin for Connector/J</td>
<td>117</td>
</tr>
<tr>
<td></td>
<td>12.3 Using the MySQL Enterprise Plugin for Connector/Net</td>
<td>121</td>
</tr>
</tbody>
</table>
Chapter 3 Installation Prerequisites

Table of Contents

3.1 Installer Files .......................................................................................................................... 19
3.2 Prerequisites ............................................................................................................................ 20
  3.2.1 System Requirements .......................................................................................................... 20
  3.2.2 Supported Platforms ........................................................................................................... 21
  3.2.3 MySQL Enterprise Monitor Repository ............................................................................... 22
3.3 Credentials Needed for Installation ......................................................................................... 23
  3.3.1 Existing Users ....................................................................................................................... 23
  3.3.2 Users Created During Installation ....................................................................................... 23
  3.3.3 Users Created on First Log-in ............................................................................................... 24

This chapter describes the process of installing the MySQL Enterprise Monitor on all operating systems.

A working installation requires the following:

- One MySQL Enterprise Service Manager. It stores its data in a database repository. You can use an existing MySQL instance for the repository, or set up a separate instance as part of the MySQL Enterprise Service Manager installation. See Chapter 4, Service Manager Installation.

- Optionally (but recommended), one or more MySQL Enterprise Monitor Agents, one for each host to monitor. See Chapter 5, Monitor Agent Installation. Install the MySQL Enterprise Service Manager first, because the Agent installation asks for credentials and network settings that you choose as you install the MySQL Enterprise Service Manager.

To minimize network overhead, you usually install the Agent on the same machine that hosts the monitored MySQL server, but you can install it on any machine that has network access to both the monitored MySQL server and the MySQL Enterprise Monitor User Interface. In other words, an agent may monitor either locally, remotely, or both.

Note

While it is possible to use a single agent to monitor multiple hosts, it is not recommended for performance reasons.

The Agent monitors the MySQL server, and transmits health and usage data back to the Service Manager. The Advisors interpret the results, which are displayed in the browser-based MySQL Enterprise Monitor User Interface.

After installing and starting the Service Manager and Agents, configure the settings in the MySQL Enterprise Monitor User Interface, as explained in Section 4.5, “MySQL Enterprise Service Manager Configuration Settings”.

3.1 Installer Files

The MySQL Enterprise Monitor files include:

- MySQL Enterprise Service Manager, MySQL Enterprise Monitor User Interface, and Advisors for the platform that you intend to execute the MySQL Enterprise Service Manager on. For a new installation, this installer is named mysqlmonitor-version-platform-installer.bin. For an upgrade installation, this installer is named mysqlmonitor-version-platform-update-installer.bin.

- One or more MySQL Enterprise Monitor Agent, one for each host. In this default scenario, the Agent installed on the same machine as a monitored MySQL instance, make a list of the platforms your
MySql servers run on, then download the Agent installer package for each of those platforms. For a new Agent installation, this installer is named `mysqlmonitoragent-version-platforminstaller.extension`. For an upgrade Agent installation, this installer is named `mysqlmonitoragent-version-platform-update-installer.extension`.

### 3.2 Prerequisites

This section describes the prerequisites for a successful MySQL Enterprise Monitor installation.

#### 3.2.1 System Requirements

This section describes the minimum and recommended system requirements for a successful MySQL Enterprise Monitor installation.

**Minimum Hardware Requirements**

This section describes the minimum hardware requirements for the Enterprise Service Monitor.

- 2 CPU Cores
- 2 GB RAM
- Disk I/O subsystem applicable to a write-intensive database

**Recommended Hardware Requirements**

This section describes the recommended hardware requirements for the Enterprise Service Manager.

- 4 CPU Cores or more
- 8 GB RAM or more
- RAID10 or RAID 0+1 disk setup

**MySQL Enterprise Monitor Disk space Requirements**

The following table lists the minimum disk space required to install the Enterprise Service Manager and Monitoring Agent for each platform.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Minimum Disk space Required by Service Manager</th>
<th>Minimum Disk space Required by Monitoring Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux x86 32-bit</td>
<td>1.1 GB</td>
<td>600 MB</td>
</tr>
<tr>
<td>Linux x86 64-bit</td>
<td>1.3 GB</td>
<td>800 MB</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>1.2 GB</td>
<td>700 MB</td>
</tr>
<tr>
<td>Solaris x86 64-bit</td>
<td>1.8 GB</td>
<td>800 MB</td>
</tr>
<tr>
<td>Solaris Sparc 64-bit</td>
<td>1.7 GB</td>
<td>600 MB</td>
</tr>
<tr>
<td>FreeBSD</td>
<td>N/A</td>
<td>300 MB (the FreeBSD installation does not include a JRE. It is assumed a compatible JRE is present on the system.)</td>
</tr>
<tr>
<td>Windows x86 32-bit</td>
<td>800 MB</td>
<td>500 MB</td>
</tr>
<tr>
<td>Windows x86 64-bit</td>
<td>800 MB</td>
<td>500 MB</td>
</tr>
</tbody>
</table>
Important

The minimum disk space values for the Monitoring Agent include the disk space required by the backlog. The backlog is used if the agent loses contact with the Service Manager and cannot transmit the collected data. The collected data is stored on the agent's local file system until communication with the Service Manager resumes. Once normal communication is resumed, the entire backlog is transmitted, then deleted from the agent's local file system.

If you choose to install the bundled MySQL Server with the Enterprise Service Manager, you must also consider the amount of disk space required by the database. This value cannot be predicted as it depends on load, number of monitored instances, and so on.

Important

If you are upgrading from a previous version of MySQL Enterprise Monitor, the upgrade process can create a full backup of all settings, including the local MySQL database used for the repository. This can result in a very large backup directory, several gigabytes in size, depending on the number of monitoring agents, and server load. Before upgrading, check the size of your existing installation and ensure you have enough disk space to run the upgrade. The upgrade also requires enough disk space for temporary files created by the upgrade process.

3.2.2 Supported Platforms

The supported platforms for MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent are listed at the following locations:

- MySQL Enterprise Service Manager Supported Platforms
- MySQL Enterprise Monitor Agent Supported Platforms

For platform support updates, see MySQL Product Support Announcements.

General Platform Recommendations

The following are recommended:

- Ensure that your Service Manager and Agent hosts are synchronized to the same time server. It is important that all times are properly synchronized.
- Ensure that your Service Manager and Agent hosts use different SSH host keys before installing.

MySQL Requirements

This section describes the MySQL Server requirements for MySQL Enterprise Monitor installation.

- The MySQL Enterprise Service Manager installation includes the latest version of MySQL Server. If you intend to use a MySQL repository other than the one bundled in the MySQL Enterprise Service Manager installation, it is recommended that you use the latest MySQL 5.6.x version.

- If you have previously configured a default login path on the same machine on which you are installing MySQL Enterprise Service Manager with the bundled repository, you must delete the cnf in which the default login details are defined before installing. If a default login path is defined, the installation fails to complete. It is recommended to install MySQL Enterprise Service Manager on a dedicated server.

- The Monitoring Agent can monitor any version of MySQL Server from version 5.0 onwards. See Section 5.2, “Creating MySQL User Accounts for the Monitor Agent” for more information on monitoring older versions.
It is not possible to monitor pre-GA versions of MySQL 5.7. That is, MySQL versions 5.7.0 to 5.7.5 are not supported. MySQL Enterprise Monitor supports monitoring of MySQL 5.7.6 onwards.

The monitoring Agent always uses `PERFORMANCE_SCHEMA.GLOBAL_STATUS` on MySQL 5.7 versions, and supports both modes of `show_compatibility_56` from MySQL 5.7.9 onwards.

To monitor versions of MySQL 5.7.8, `show_compatibility_56` must be set to OFF.

### 3.2.3 MySQL Enterprise Monitor Repository

The Enterprise Service Manager requires a repository to store its data. The installer optionally installs a local, clean repository for this purpose. However, you can choose not to install the bundled MySQL Server and use another repository instead. This repository can be on the same machine as the Enterprise Service Manager, or on a remote machine.

It is strongly recommended that you use the bundled MySQL instance as the MySQL Enterprise Monitor repository. Only use an external repository if you have a compelling business reason for doing so.

The bundled MySQL instance has been comprehensively tested and tuned for use with the MySQL Enterprise Service Manager.

The MySQL Enterprise Monitor upgrade installer can only upgrade a bundled MySQL, not an external one.

The various scripts delivered with MySQL Enterprise Service Manager only work with the bundled MySQL.

The repository instance must be present before starting the MySQL Enterprise Monitor installation.

It is strongly recommended you use a clean installation of MySQL Server as the Enterprise Service Manager repository and do not use this server for any other purpose.

You must make several configuration changes to enable it for use as the repository.

Ensure the following:

- The MySQL Server version is 5.6.14 or higher, or 5.7.9, or higher.

- The InnoDB storage engine is available.

- SSL is enabled. For more information on configuring SSL for MySQL Enterprise Monitor, see Section 6.2, “Installing SSL Certificates”.

It is not possible to use MySQL 5.1.x, or 5.5.x, for the MySQL Enterprise Monitor repository.
You must ensure the following in the MySQL Server configuration:

- Query Cache must not be enabled.
- Set `innodb_file_per_table=1`.
- Set `innodb_file_format=Barracuda`.
- On Linux/Unix hosts, ensure `innodb_flush_method=O_Direct`, except on Solaris if ZFS is used. If using ZFS, comment out this parameter.
- It is recommended to set `innodb_log_file_size=2048M`.
- Define a Service Manager user to enable the MySQL Enterprise Service Manager to connect to, and modify, the repository. This user must have the following privileges:
  - All privileges on `mem%.*` tables
  - `CREATE` and `INSERT` on `mysql.inventory`
  - `REPLICATION CLIENT`, `SUPER`, `PROCESS`, and `SELECT` on all databases in the repository.

The Service Manager user's credentials are required by the MySQL Enterprise Service Manager installation process.

### 3.3 Credentials Needed for Installation

Before installing the MySQL Enterprise Monitor components, gather credentials (a root user ID and password) for all the MySQL servers you plan to monitor. The Agent installation requires a dedicated user ID in each monitored MySQL server, and optional limited and general users that the Installer can create for you.

**Note**

With MySQL 5.5.16 and higher, you can configure these user IDs to authenticate using the PAM Authentication plugin. Currently, MySQL Enterprise Monitor does not support authentication through the Windows Native Authentication plugin.

Optionally, gather credentials for your My Oracle Support account, which you can specify in the MySQL Enterprise Monitor User Interface Settings tab.

The following sections outline the users associated with the MySQL Enterprise Monitor.

#### 3.3.1 Existing Users

The **MySQL user**: For Monitor Agents to report the status of a MySQL server, they connect to a MySQL user with privileges to read any data on that server: `SHOW DATABASES`, `REPLICATION CLIENT`, `SUPER`, `CREATE`, and `SELECT`. If you already have such a user on a MySQL server, specify its credentials when installing the Agent for that server. For details about this account, see Section 5.2, “Creating MySQL User Accounts for the Monitor Agent”.

The **My Oracle Support user**: These are the credentials you use to log in to the My Oracle Support web site. The What's New page accesses this account to receive updates and examine relevant service issues.

#### 3.3.2 Users 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 GUI program such as MySQL Workbench.

At the end of MySQL Enterprise Service Manager installation, the file configuration_report.txt is created, and along with other configuration details, contains some of the credentials of the repository manager. 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

### 3.3.3 Users Created on First Log-in

The **Manager user**: This user is the administrator of the MySQL Enterprise Monitor User Interface. The first time you log in to the Monitor UI, log in as this user. You choose the ID and 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 Monitor UI. You choose the ID and password for this user.

**Note**

The Monitor Agent communicates both with the MySQL Enterprise Monitor User Interface, and with the MySQL server it is monitoring. For a description of the agent as a MySQL user, see Section 3.3.1, “Existing Users”.
Chapter 4 Service Manager Installation

Table of Contents

4.1 MySQL Enterprise Monitor Installation Types ................................................................. 25
4.2 MySQL Enterprise Service Manager Graphical Installation Wizard ................................ 26
4.3 Text-Based Installation .................................................................................................... 29
4.4 Starting/Stopping the MySQL Enterprise Monitor Services ........................................... 29
4.5 MySQL Enterprise Service Manager Configuration Settings .......................................... 30
  4.5.1 Initial Monitor UI Log-In .............................................................................................. 31
  4.5.2 Setting the Timezone and Locale ................................................................................ 32

This chapter describes the installation of the MySQL Enterprise Service Manager.

Important

Due to changes in TLS support, as of MySQL Enterprise Monitor 3.0.22, it is not possible for the MySQL Enterprise Service Manager to communicate with earlier versions of the MySQL Enterprise Monitor Agent.

MySQL Enterprise Service Manager 3.0.22 cannot communicate with any MySQL Enterprise Monitor Agent earlier than version 3.0.22.

The MySQL Enterprise Service Manager installer installs the following components:

- Apache Tomcat: mandatory component. Servlet container and web server which hosts the MySQL Enterprise Service Manager.

- Java Runtime Environment (JRE): mandatory component. Required by Tomcat.

- MySQL Server: optional component. Used to store the data from the monitored hosts and instances. Referred to, throughout this document, as the repository. It is also possible to use another MySQL instance as the repository.

4.1 MySQL Enterprise Monitor Installation Types

The MySQL Enterprise Service Manager installer enables you to choose your installation type. This choice sets parameters which suit your installation type.

The following are the possible installation types:

- Small: 1 to 5 MySQL Servers monitored from a laptop or low-end server with no more than 4GB of RAM.

Important

Under certain circumstances, the small installation type can result in Out Of Memory Errors. At the time of writing, it is recommended you use the Medium installation instead of the Small.

- Medium: Up to 100 MySQL Servers monitored from a medium-sized, but shared, server with 4 to 8GB of RAM.

- Large: More than 100 MySQL Servers monitored from a high-end server, dedicated to MySQL Enterprise Service Manager, with more than 8GB RAM.
These parameters are set in the following configuration files:

- **setenv.sh/setenv.bat**:
  - Tomcat Heap Size (`-Xms` and `-Xmx`): defines the minimum (`-Xms`) and maximum (`-Xmx`) amount of RAM available to Tomcat's JVM. `-Xmx` and `-Xms` are set to the same value.
  - Tomcat MaxPermSize: defines the maximum size of the pool containing the data used by Tomcat's JVM.

- **my.cnf/my.ini**:
  - `table_definition_cache`: defines the number of table definitions that can be stored in the definition cache.
  - `innodb_buffer_pool_size`: defines the size, in megabytes, of the InnoDB buffer pool.

### Table 4.1 Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tomcat Heap Size</td>
<td>256MB</td>
<td>768MB</td>
<td>2048MB</td>
</tr>
<tr>
<td>Tomcat MaxPermSize</td>
<td>200MB</td>
<td>512MB</td>
<td>1024MB</td>
</tr>
<tr>
<td><code>table_definition_cache</code></td>
<td>800</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td><code>innodb_buffer_pool_size</code></td>
<td>100MB</td>
<td>768MB</td>
<td>8096MB</td>
</tr>
</tbody>
</table>

**Important**

These values are not hard-coded. You can change them, if your installation requires it, by editing `setenv.sh/setenv.bat`, or `my.cnf/my.ini`.

For more information, see Chapter 9, *Performance Tuning MySQL Enterprise Monitor*.

### 4.2 MySQL Enterprise Service Manager Graphical Installation Wizard

This section describes how to install the MySQL Enterprise Service Manager using the Installation Wizard. This process is identical across all supported platforms.

**Note**

On UNIX and Linux platforms, ensure the installer is executable before you begin.

**Important**

It is recommended to install MySQL Enterprise Service Manager as root, but not to run MySQL Enterprise Service Manager as root. If you install as root, you are prompted to create a user for MySQL Enterprise Service Manager. If you do not install as root, MySQL Enterprise Service Manager cannot start automatically on system boot and you must be started manually.

To install MySQL Enterprise Service Manager, do the following:

1. Run the installer as required by the operating system.
2. The language selection dialog is displayed. Choose a language and click **OK**.

The following information is displayed:
3. Click **OK** to continue.

4. On the **Welcome** dialog, click **Forward**.

   The **Installation Directory** dialog is displayed.

5. para Change the installation directory or accept the default path and click **Forward**.

   The **Select Requirements** dialog is displayed.

6. Select the size of installation required. For more information, see Section 4.1, “MySQL Enterprise Monitor Installation Types”.

   Click **Forward**.

   The **Tomcat Server Options** dialog is displayed.

7. Complete the following fields as required:

   - **Tomcat Server Port**: Default value is 18080. This port is required by the upgrade from version 2.3 to 3.0, only. It enables the 2.3 Agents to communicate with MySQL Enterprise Service Manager 3.0. 2.3 Agents did not support SSL.

     _Note_
     
     If you are performing a clean installation of 3.0, and no 2.3 Agents are present, clear this field.

   - **Tomcat SSL Port**: Default value is 18443. This port is mandatory for communication with 3.0 Agents, which must use SSL to communicate with the MySQL Enterprise Service Manager.

     Click **Forward**.

   The **Service Manager User Account** dialog is displayed.

8. Enter the name of the user account MySQL Enterprise Service Manager will run under. If this user account does not exist, it is created by the installer.

   Click **Forward**.

   The **Database Installation** dialog is displayed.

9. Select one of the following options:

   - **I wish to use the bundled MySQL database**: select to install a MySQL server.

     _Important_
     
     If you choose the bundled server option, the Service Manager user defined by the installation procedure is granted complete control of the repository. This is done using `GRANT ALL PRIVILEGES ON * TO 'SM_UserName'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;`.

   - **I wish to use an existing MySQL database**: select to use an existing MySQL server as the repository.
Important

If you choose the existing server option, you must ensure the prerequisites listed in Section 3.2.3, “MySQL Enterprise Monitor Repository” are met before installing MySQL Enterprise Service Manager.

Click Forward.

The Repository Configuration dialog is displayed.

10. Complete the following fields:

- **Repository Username**: enter the username used by MySQL Enterprise Service Manager to connect to the repository. If you chose to use an existing database, this user must already exist on the target MySQL instance.

  The default username is `service_manager`.

- **Password/Re-enter**: enter the password and confirm in the Re-enter field.

- **MySQL Hostname or IP address**: (Displayed if you chose to use an existing MySQL database, only) enter the hostname or IP address of the MySQL instance.

- **MySQL Database Port**: enter the port MySQL Enterprise Service Manager uses to connect to the MySQL instance. If you chose the bundled repository, the default port number is 13306. If you chose to use an existing instance, the default port number is 3306.

- **MySQL Database Name**: enter the name of the MySQL Enterprise Service Manager repository. This is useful if you intend to use multiple MySQL Enterprise Service Manager installations, but want to host their repositories on a single MySQL server. Each MySQL Enterprise Service Manager must have a uniquely named repository. It is not possible for MySQL Enterprise Service Managers to share a repository.

- **Use SSL when connecting to the database**: enables SSL encryption for all communication between MySQL Enterprise Service Manager and the repository.

- On Mac OS X platforms, you are prompted to optionally install MySQL Enterprise Service Manager as a service. This setting enables MySQL Enterprise Service Manager to start when the machine is started. You must provide the Administrator password to install MySQL Enterprise Service Manager as a service.

Click Forward.

The Configuration Report dialog is displayed.

11. Click Forward to install MySQL Enterprise Service Manager.

**Installation Log**

The installation log file is written to the root of the installation directory.

The installation log uses the following naming convention: `install.log`.

The log file records all files installed and all actions taken by the installer, such as starting services, filling database tables, and so on. A similar log file is also created by the uninstall process.

If the installation is upgraded, the existing installation log is backed up to the backup directory and replaced by the installation log for the upgrade.
4.3 Text-Based Installation

The steps and options of the text-based installation are identical to those described in Section 4.2, “MySQL Enterprise Service Manager Graphical Installation Wizard”.

**Note**
There is no text-mode installation available for Microsoft Windows platforms.

To start the text-based installer, do the following:

1. Run the installer with the following option:

   ```
   --mode text
   ```

   The following example shows how to start the text-mode installation on a 64-bit Linux system:

   ```
   shell>./mysql-monitor-3.0.18.3095-linux-x86-64bit-installer.bin --mode text
   ```

   The text installation process starts.

2. Follow the instructions onscreen. The options and values are identical to those described in Section 11.4, “Graphical Installation Wizard”.

   After the Service Manager is installed, you can configure the MySQL Enterprise Monitor User Interface, as explained in Section 4.5, “MySQL Enterprise Service Manager Configuration Settings”.

4.4 Starting/Stopping the MySQL Enterprise Monitor Services

This section describes how to control the MySQL Enterprise Service Manager services on UNIX, Linux and Mac platforms. Microsoft Windows supports several additional methods, which are described in Starting/Stopping the MySQL Enterprise Monitor Services on Windows.

The following services are installed by MySQL Enterprise Service Manager:

- MySQL Server
- Tomcat Server.

Access the MySQL Enterprise Service Manager services using the script `mysqlmonitorctl.sh/mysqlmonitor.bat` which is installed in the root of your MySQL Enterprise Service Manager installation directory. To see the available options, run the command `mysqlmonitorctl.sh help`.

The `help` parameter produces the following output:

```
usage: ./mysqlmonitorctl.sh help
          ./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
```

To autostart all the Service Manager components, call the `mysqlmonitorctl.sh start` from your start-up script.

To start the service:
Starting/Stopping the MySQL Enterprise Monitor Services on Windows

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

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

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

**Important**

The Service Manager can take some time to start and become usable after **mysqlmonitorctl.sh start** completes.

This script can also 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

After the Service Manager is installed, you can configure the MySQL Enterprise Monitor User Interface, as explained in Section 4.5, “MySQL Enterprise Service Manager Configuration Settings”.

Starting/Stopping the MySQL Enterprise Monitor Services on Windows

You can stop or start the MySQL Enterprise Service Manager services in the following additional ways:

- The **Start/Stop** MySQL Enterprise Monitor Services items on the Windows Start menu.
- The **Services** pane of the **Microsoft Management Console**. Right-click on the either of the **MySQL Enterprise** services to display the available options.
- The Windows command line, using the **sc** or **net** commands.

From the command line, the service names are **mysqlenterprisetomcat** and **mysqlenterprisemysql**.

For example:

```shell
sc start mysqlenterprisemysql
```

**Note**

The command line terminal must be started using the **Run as Administrator** option.

4.5 MySQL Enterprise Service Manager Configuration Settings

The MySQL Enterprise Monitor User Interface is the web-based interface to the Service Manager. The procedure for starting the Monitor UI is identical for all platforms.

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

Otherwise, you can view the MySQL Enterprise Monitor User Interface by typing **https://localhost:18443/** ("18443" is the default port number, adjust accordingly if you altered this
configuration), into the address bar of your web browser. To see the host name and port to use, check the configuration_report.txt file.

Under Microsoft Windows, you can also open the Monitor UI by choosing the MySQL menu item and finding the MySQL Enterprise Monitor entry. Under this entry, choose Start Service Manager.

**Important**

On first start, MySQL Enterprise Service Manager can take some time to start while the services and database initialize.

### 4.5.1 Initial Monitor UI Log-In

If this is the first time logging in to the Monitor UI, the following page is displayed:

**Figure 4.1 Initial setup for the MySQL Enterprise Monitor User Interface**

Use this page to perform the following tasks:

- Create a user name and password for the **Manager User**. This is for logging into and viewing the MySQL Enterprise Monitor User Interface, as well as administering the configuration of the MySQL Enterprise Service Manager, such as adding new users and configuring settings.

- Create a user name and password for the MySQL Enterprise Monitor Agent. This is a user that each Agent can use to login to the MySQL Enterprise Service Manager.

- Configure your **Data Purging Behavior** preferences.

  Although these settings control the amount of disk space used, changing them later to lower values may not reclaim disk space automatically, as you would have to dump-and-reload the table, and InnoDB tables never shrink.

- Configure your preferences for **Online Updates**.

  In the **Create Manager User** section of this screen, enter credentials for the Monitor UI administrator. This creates the Manager user described in Section 3.3.3, “Users Created on First Log-in”. Record the user name and password, as these credentials are required for any future login.
In the Create Agent User section of this screen, enter the credentials for the agent. This is the user described in Section 3.3.3, “Users Created on First Log-in”. The agent must log in to report its findings. Record the agent's credentials; this information is required when installing the agent.

After specifying all settings, click the Complete Setup button. When you log in, a message reports that the Advisors are now scheduled.

4.5.2 Setting the Timezone and Locale

If this is the first time that you have launched the MySQL Enterprise Monitor User Interface, 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

Make sure to set the time zone (and the system clock) correctly because this setting affects the way the graphs display. To change the time zone or locale, see Section 15.4, “User Preferences”.

The locale chosen determines the default language displayed when logging in to the Monitor UI. This overrides the default browser settings whenever this specific user logs in.

After specifying your time zone and locale, the Monitor UI opens on the What’s New page.

At this point the MySQL Enterprise Service Manager Repository is being monitored, the built-in Agent is attempting to auto-discover additional MySQL instances on the host, and you are now ready to follow the Chapter 6, Post-installation Considerations guide.
Chapter 5 Monitor Agent Installation

Table of Contents

5.1 General Agent Related Notes ................................................................. 33
5.2 Creating MySQL User Accounts for the Monitor Agent ................................ 34
5.3 Java Considerations on Linux .................................................................. 37
5.4 Installing the Monitor Agent on Unix and Linux ........................................ 37
5.5 Installing the Agent on Microsoft Windows ................................................. 42
5.6 Installing the Agent on Mac OS X ........................................................... 45
5.7 Starting/Stopping the MySQL Enterprise Monitor Agent ............................. 52
  5.7.1 Starting/Stopping the Agent on Windows ............................................. 53
  5.7.2 Starting/Stopping the Agent on Mac OS X .......................................... 54
  5.7.3 Starting/Stopping the Agent on Unix ................................................... 55
  5.7.4 sql_mode .......................................................................................... 56
5.8 Monitoring Multiple MySQL Servers ....................................................... 56
5.9 Configuring an Agent to Monitor a Remote MySQL Server ........................... 56
5.10 Monitoring Outside the Firewall with an SSH Tunnel ................................. 57
5.11 HTTP Connection Timeout ..................................................................... 58
5.12 Troubleshooting the Agent ....................................................................... 58
5.13 Agent Backlog ....................................................................................... 59

A MySQL Enterprise Monitor Agent monitors a MySQL server and sends data to the MySQL Enterprise Service Manager. The data is interpreted by the MySQL Enterprise Advisors and displayed in the MySQL Enterprise Monitor User Interface. The following section describes how to install the Agent on all platforms.

Important

Due to changes in TLS support, as of MySQL Enterprise Monitor 3.0.22, it is not possible for the MySQL Enterprise Service Manager to communicate with earlier versions of the MySQL Enterprise Monitor Agent.

MySQL Enterprise Service Manager 3.0.22 can communicate with MySQL Enterprise Monitor Agent 3.0.22, only.

5.1 General Agent Related Notes

This section describes important features of the Agent and describes how the 3.0.x Agent differs from the 2.3.x Agent.

• The Agent uses three users with different connection levels: Admin, General (optional), and Limited (optional). These can be created manually or by the installation and configuration process.

• Typically, one Agent is installed per host, and the Agent monitors the host and all MySQL instances on it. An Agent may also monitor remote MySQL instances.

• Agents automatically detect MySQL instances on a host. Adding the new MySQL instance may be performed in the MySQL Enterprise Monitor UI or from the command line.

• Choosing a MySQL Instance to monitor during the installation is optional. If you choose to define a MySQL Instance while running the Installer, additional MySQL Instances on the host are detected and reported in the MySQL Enterprise Monitor User Interface. From there, you can add the appropriate configuration information.
3.0 Agent differences from 2.3

- In order to properly detect a local connection in IPv6, the Agent requires that forward resolution exists on the system from localhost to `::1`, which could mean editing the `/etc/hosts` configuration file.

  This is because the "SHOW PROCESSLIST" statement always reports "localhost" even when bound to `::1` without an address resolution. If localhost resolution is not configured for IPv6, the Agent cannot detect a local IPv6 MySQL server connection, even when it occurs.

3.0 Agent differences from 2.3

Note

The Proxy and Aggregator are not included in the MySQL Enterprise Monitor Agent v3.0 installer. As of version 3.0.14, the Proxy and Aggregator have a dedicated installation package. For more information, see Chapter 11, Proxy and Aggregator Installation.

- Multiple connection levels (General, Limited, Admin) are now defined and utilized by the Agent. For more information, see Section 5.2, "Creating MySQL User Accounts for the Monitor Agent".

- You may assign a monitored MySQL instance to a group via the Agent installer, which is displayed in the MySQL Enterprise Service Manager.

- The old Agent's configuration files (`mysql-monitor-agent.ini` and `agent-instance.ini`) no longer exist. Use `custom.xml` instead.

- Passwords are now stored in an encrypted format, so you can no longer recover passwords by looking in the configuration files.

- The Service Manager now bundles an Agent, which monitors the host on which it is installed, scans for all MySQL instances on the host, and also monitors the Service Manager repository database.

Note

It is recommended to install MySQL Enterprise Service Manager on a dedicated server with no other MySQL instances installed.

- For a list of supported platforms that the Agent installation supports, see http://www.mysql.com/support/supportedplatforms/enterprise-monitor.html.

5.2 Creating MySQL User Accounts for the Monitor Agent

The MySQL Enterprise Monitor Agent requires a user configured within each MySQL instance that is being monitored with suitable privileges to collect information about the server, including variable names, replication, and storage engine status information.

The Agent requires the Admin user, and can optionally use General or Limited users, or both, depending on the system's security requirements. During the installation process, you are prompted to create General and Limited users. You can allow the agent to connect to the database using the Admin user for all tasks but it is recommended to create the General or Limited users for tasks which do not require root access to the database. It is not necessary to create both users. It is possible to create one or the other. The Agent uses the user with the lowest, required privileges for the query and changes to a user with higher privileges only if the query requires it.

Important

All configured users are checked to ensure they have the correct privileges. If they do not have the required privileges, the connection is regarded as a bad connection and added to the Bad Connections list on the MySQL Instances dashboard.
Creating the Admin User

- **Admin:** Typically this is the ‘root’ user, otherwise it a user that has the **SUPER** privilege, and has **CREATE** and **INSERT** privileges on the schema the inventory table will be created on (the inventory table stores unique identifiers for the MySQL instance, and is created in the **mysql** schema by default). The **SUPER** privilege is required to temporarily switch off replication when creating and populating the inventory table, as well as running certain statements such as **SHOW MASTER LOGS** or **SHOW ENGINE INNODB STATUS**, depending on the version that is being monitored.

If you choose to **Auto-Create Less Privileged Users**, this Administrative user also creates those with the required privileges to monitor this instance. In this case, this user should also have the **CREATE USER**, **PROCESS**, **REPLICATION_CLIENT**, **SELECT**, **SHOW DATABASES** privileges globally **WITH GRANT OPTION**, and **UPDATE** on the **performance_schema.tables** table.

- **General:** This optional user handles general monitoring tasks that do not require **SUPER** level privileges. Lower privileged users are used until higher privileges are required. In which case, MEM temporarily logs in as the SUPER privileged user, and then falls back to the general user.

If you are manually managing this user, it should have at least the **PROCESS**, **REPLICATION_CLIENT**, **SELECT**, and **SHOW DATABASES** privileges globally, and **UPDATE** on the **performance_schema.tables** table. If you intend to use EXPLAIN on views, you must also grant **SHOW VIEW**.

---

**Important**

If you are monitoring MySQL 5.1.63, or earlier, you must grant the **SUPER** privilege to the General user. The agent requires this privilege to use the **SHOW BINARY LOGS** statement on the monitored instance.

- **Limited:** This optional user is used for statements that should be limited to a single connection.

Examples of these types of statements include getting database metadata from **INFORMATION_SCHEMA** tables (which with large numbers of databases and tables can become costly), or any custom SQL that is used to monitor application specific statistics.

If you are manually managing this user, it should have at least the **SELECT** and **SHOW DATABASES** privileges globally, and **UPDATE** on the **performance_schema.tables** table. If you intend to use EXPLAIN on views, you must also grant **SHOW VIEW**.

Creating the Admin User

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

- **SHOW DATABASES:** The MySQL Enterprise Monitor Agent can gather inventory about the monitored MySQL server.
- **REPLICATION CLIENT:** The MySQL Enterprise Monitor Agent can gather Replication master/slave status data. This privilege is only needed if you use the MySQL Replication Advisor Rules.
- **SELECT:** The MySQL Enterprise Monitor Agent can collect statistics for table objects.
- **SUPER:** The MySQL Enterprise Monitor Agent can execute **SHOW ENGINE INNODB STATUS** to collect data about InnoDB tables. This privilege is also required to obtain replication information using **SHOW MASTER STATUS**, and to temporarily switch off replication when populating the **mysql.inventory** table used to identify the MySQL instance.
- **PROCESS:** When monitoring a MySQL server running MySQL 5.1.24 or above with **InnoDB**, the **PROCESS** privilege is required to execute **SHOW ENGINE INNODB STATUS**.
- **INSERT:** Required to create the UUID required by the agent.
Creating the Admin User

- **CREATE**: The MySQL Enterprise Monitor Agent can create tables. During discovery, the agent creates the table `inventory` within the `mysql` database that stores the UUID for the server. Without this table, the agent cannot determine the UUID of the server, which it sends along with other information to MySQL Enterprise Service Manager.

- **UPDATE** on the `performance_schema.threads` table. This is done to prevent **SQL Statement Generates Warnings or Errors** events which can be triggered by EXPLAIN plans run by the Query Analyzer. These warnings are generated because the `Performance_Schema` captures only 1024 characters of each query. Granting this privilege enables the connection to `Performance_Schema` to be dropped before the EXPLAIN and reconnected after the EXPLAIN finishes.

  **Note**
  
  If you manage your General and Limited users manually, you must also grant this privilege to those users.

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

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

  **Note**
  
  When using **Auto-Create Less Privileged Users**, also add `WITH GRANT OPTION` to the above query.

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

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

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

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

  **Note**
  
  Perform this step **after** after running the agent on the corresponding MySQL server to ensure that the `mysql.inventory` table is created correctly. Run the agent, shut the agent down, run the above `GRANT` statement, and then restart the agent.

  If the agent cannot access the information from the table, a warning containing this information is written to the agent log.

  **Note**
  
  You might disable logging for the grant statement to prevent the grant information being replicated to the slaves. In this case, execute the statement `SET SQL_LOG_BIN=0` before executing the above `GRANT` statement.
Creating the Limited and General Users

If the Admin user has the necessary privileges to create other users, you can check the Auto-Create Less Privileged Users checkbox, enter credentials for those users, and they are created for you.

If the Auto-Create Less Privileged Users box is unchecked and the credentials for the General and Limited users blank, the Agent only uses the Admin user for monitoring.

If the Auto-Create Less Privileged Users box is unchecked, you can enter credentials for the General and Limited users. If you define these users, you must create them on the monitored assets manually. The installer attempts to validate these users and displays a warning message if they are invalid. The installation process continues, and the Agent works properly, but you must create those users later.

In a typical configuration, the Agent runs on the same host as the MySQL server it is monitoring, so the host name is often localhost. If the Agent is running on a machine other than the monitored MySQL server(s), then change localhost to the appropriate value. For more information about remote monitoring, see Section 5.9, “Configuring an Agent to Monitor a Remote MySQL Server”.

5.3 Java Considerations on Linux

The MySQL Enterprise Monitor Agent installers and updaters for UNIX-based platforms are delivered with and without a compatible JVM. For those installers which do not include a compatible JVM, you must download and install a compatible version if you do not already have one installed. Consult your platform’s support documentation for information on appropriate installations.

Important

On 64-bit platforms, it is recommended to use a 32-bit JRE with the 32-bit MySQL Enterprise Monitor Agent. The 32-bit version uses considerably less RAM than the 64-bit version. For more information, see Compatibility Libraries.

Compatibility Libraries

If you intend to use a 32-bit JVM on a 64-bit platform, ensure that you have the correct compatibility libraries installed, enabling the 64-bit application to run with a 32-bit JVM.

These libraries differ between Linux versions. For example, on Debian or Ubuntu, you must ensure Multiarch is installed or, if using earlier versions, ia32-libs. On RedHat, or CentOS, you must ensure the glibc.i686, libXext.i686 and libXtst.i686 libraries are installed. Consult your platform documentation for more information on compatibility.

5.4 Installing the Monitor Agent on Unix and Linux

Note

To install the MySQL Enterprise Monitor Agent on Linux systems, you must have the Linux Standards Base (LSB) initialization functions installed. To check the existence of the LSB components, look for an LSB package within your Linux package management environment. For example, on RedHat and other RPM-based distributions:

```
shell> rpm -qa | grep -i lsb
redhat-lsb-3.1-19.fc8.x86_64
```

Under Debian/Ubuntu:

```
shell> dpkg -l|grep -i lsb
ii lsb-base 3.2-20ubuntu4
ii lsb-release 3.2-20ubuntu4
```
Installing the Monitor Agent on Unix and Linux

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).

**Important**
For performance reasons, it is strongly recommended that you install the 32-bit agent and JVM package on 64-bit platforms. For more information, see Compatibility Libraries.

Ensure that this file is executable by typing:

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

**Note**
To install to the default directory (`/opt/mysql/enterprise/agent`), log in as `root` first. Installing as an unprivileged user installs to the `/home/user_name/mysql/enterprise/agent` directory.

To automatically start the agent upon rebooting, you must install while logged in as `root`. If you install as an unprivileged user, you must start the agent yourself after each reboot.

**Note**
If MySQL Enterprise Monitor Agent is installed as the root user, directories and files that the Agent writes to are owned by the `mysql` user in the `mysql` group, which includes `logs/`, `spool/`, and `etc/agentManaged`. The Agent is started by, and runs as, the `mysql` user.

The following section covers installation from the command line. You can install the Monitor Agent graphically by running the installer from within a windows manager. In both cases, the steps are identical.

You can also install the Monitor Agent in unattended mode, for example to do multiple installations. For more information on unattended installation, see Section 8.1, "Unattended Installation".

**Note**
To install multiple agents on the same machine, use the `agentservicename` option with the installer to set a unique service name each time. For more information, see `installer_agentservicename`.

**Note**
On FreeBSD, the Agent Installer does not bundle the required JRE 8.

Begin installation from the command line by typing:

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

The following section shows the various options. Default values are indicated by square brackets; to select them, press Enter. Otherwise, enter a value of your choosing.

1. First, select the Language you want to use during the installation process:
Installing the Monitor Agent on Unix and Linux

Language Selection
Please select the installation language
[1] English - English
Please choose an option [1] :

2. Next, specify the directory in which to install the agent:

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

3. If the installer identifies that an existing version of the MySQL Enterprise Monitor Agent is installed, provide an alternative name for the installation. This defines the name of the file created in /etc/init.d used to start the agent. You can enter a different name to perform multiple installations.

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

How will the agent connect to the database it is monitoring?
[1] TCP/IP
[2] Socket
Please choose an option [1] :

5. Choose whether or not you want to define a MySQL instance to monitor, or wait and configure it in the GUI.

Monitoring Options
You can configure the Agent to monitor this host (file systems, CPU, RAM, etc.) and then use the Monitor UI to furnish connection parameters for all current and future running MySQL Instances. This can be automated or done manually for each MySQL Instance discovered by the Agent. (Note: scanning for running MySQL processes is not available on Windows, but you can manually add new connections and parameters from the Monitor UI as well.)

Monitoring options:
[1] Host only: Configure the Agent to monitor this host and then use the Monitor UI to furnish connection parameters for current and future running MySQL Instances.
[2] Host and database: Configure the Agent to monitor this host and furnish connection parameters for a specific MySQL Instance now. This process may be scripted. Once installed, this Agent will also continuously look for new MySQL Instances to monitor as described above.
Please choose an option [2] :

6. The installer copies the necessary files:

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

39
Please wait while Setup installs MySQL Enterprise Monitor Agent on your computer.

<table>
<thead>
<tr>
<th>Installing</th>
<th>0%</th>
<th>50%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Next, provide the **Agent** user that you created when installing the MySQL Enterprise Service Manager. This user connects to the Service Manager.

MySQL Enterprise Monitor UI Options

<table>
<thead>
<tr>
<th>Hostname or IP address [127.0.0.1]:</th>
<th>127.0.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tomcat SSL Port [18443]:</td>
<td>18443</td>
</tr>
<tr>
<td>Agent Username [agent]:</td>
<td>agent</td>
</tr>
<tr>
<td>Agent Password :</td>
<td></td>
</tr>
<tr>
<td>Re-enter:</td>
<td></td>
</tr>
</tbody>
</table>

8. If you chose to monitor a MySQL server, next you provide the connection options for communicating with the server. This is a MySQL instance that you are configuring to monitor, and is typically on the host that the Agent is being installed on.

If you selected TCP/IP, you enter the TCP/IP address and port number:

MySQL hostname or IP address [127.0.0.1]:

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

MySQL Port [3306]:

Admin User [root]:

Admin Password :

Re-enter Password :

Monitor UI Group []:

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

### Note

The monitor agent always associates “localhost” with the TCP/IP address 127.0.0.1, not the MySQL socket. This is in contrast to the MySQL Command Line Tool, which connects via the MySQL socket by default on Unix, if the hostname “localhost” is specified.

If the MySQL server to monitor was started with the `--skip-networking` option, then you cannot connect to it via TCP/IP, as the server does not listen for TCP/IP connections. In this case, configure the monitor agent to use the MySQL socket. During installation, select “socket” rather than “TCP/IP” and then specify the MySQL socket name. Or, after installation, reconfigure this using the `agent.sh` configuration script. For further information, refer to Section D.2, “MySQL Enterprise Monitor Agent Reference”.

---

| Note | The monitor agent always associates “localhost” with the TCP/IP address 127.0.0.1, not the MySQL socket. This is in contrast to the MySQL Command Line Tool, which connects via the MySQL socket by default on Unix, if the hostname “localhost” is specified. If the MySQL server to monitor was started with the `--skip-networking` option, then you cannot connect to it via TCP/IP, as the server does not listen for TCP/IP connections. In this case, configure the monitor agent to use the MySQL socket. During installation, select “socket” rather than “TCP/IP” and then specify the MySQL socket name. Or, after installation, reconfigure this using the `agent.sh` configuration script. For further information, refer to Section D.2, “MySQL Enterprise Monitor Agent Reference”. |
If the MySQL server to be monitored was started using the command option `--bind-address`, then the server only listens for connections on the IP address specified, that is, the IP address of the MySQL server. If the monitor agent was started using TCP/IP networking and the default address of 127.0.0.1, it cannot connect to the server to be monitored. Also, if “localhost” is specified as the host name during agent configuration, a connection is not established, as the server listens for connections on the address specified with the `--bind-address` option, not 127.0.0.1.

9. If you want to create the less privileged Limited and General user accounts on the MySQL instance you are configuring to monitor, then select [Y]:

```
Agent User Account Creation

IMPORTANT: Please see the User Guide for more information about Agent user accounts.

Auto-Create Less Privileged Users [Y/n]:
```

10. If you decided to create the less privileged users, then define their usernames and passwords:

```
Auto-Create Less Privileged Users [Y/n]: Y

General Username []: general
General Password :
Re-enter Password :
Limited Username []: limited
Limited Password :
Re-enter Password :
```

11. A Configuration Report is now displayed.

```
Configuration Report

MySQL Enterprise Monitor Agent (Version 3.0.0.2878 : 3.0.0.2878)

The settings you specified are listed below.

Note that if you are using a Connector to collect Query Analyzer data, you will need some of these settings to configure the Connector. See the manual for more information.

Installation directory: /opt/mysql/enterprise/agent

MySQL Enterprise Monitor UI:
-------------------------
Hostname or IP address: 127.0.0.1
Tomcat Server Port: 18443
Use SSL: yes

Monitored MySQL Database:
-------------------------
Hostname or IP address: 127.0.0.1
```

12. Before installation starts, you receive a summary of the installation settings that you specified:

```
Here are the settings you specified:

Installation directory: /opt/mysql/enterprise/agent
```
Installing the Agent on Microsoft Windows

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

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

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

Start MySQL Enterprise Monitor Agent

The MySQL Enterprise Monitor Agent was successfully installed. To start the Agent please invoke:

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

Press [Enter] to continue :

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

13. Finally, you can read the supplied README file. The README is the share/doc/README_en.txt file within the agent installation directory.

For information on starting the agent, see Section 5.7.3, “Starting/Stopping the Agent on Unix”.

5.5 Installing the Agent on Microsoft Windows

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

**Note**
To install the agent as a Windows service, you must run the installation as a privileged user.

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

You can install the Monitor Agent in unattended mode, for example, to do multiple installations. For more information on this topic, see Section 8.1, “Unattended Installation”.

**Note**
To install multiple agents on the same machine, use the agentservicename option with the installer to install each agent with a unique service name. For more information, see installer_agentservicename If the installer...
Installing the Agent on Microsoft Windows

identifies an existing installation and you do not specify an alternative service name, the installer stops.

The Agent Installer for Microsoft Windows is only built using the 32-bit architecture. Installing a 32-bit Agent on a 64-bit version of Microsoft Windows is expected, although the default path will include "Program Files (x86)" instead of "Program Files". For website usability reasons, an Agent Installer binary may list a "-64-" in the filename, but note that it is a 32-bit build.

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

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.

4. Choose either **Host only** or **Host and database**. You can set up MySQL instances for monitoring from the MySQL Enterprise Monitor User Interface, and also define a MySQL instance to start monitoring right away.

**Figure 5.1 Installing Agent on Windows: Host options**

![Image of setup options](image)

**Note**
The Auto-discovery process, which automatically scans a system for MySQL instances to monitor, does not function on Microsoft Windows. You must configure them manually, either via the command-line or from the MySQL Enterprise Monitor User Interface.

5. The host name, port and Agent authentication information must be entered. Enter the required information, as defined when installing the MySQL Enterprise Service Manager, and then click **Next** to continue.

**Note**
This is the user that connects the Agent to the Service Manager. It is defined by the Service Manager, and Agent users can be modified under **Settings**, **Manage Users**. Their role is defined as "agent".
Installing the Agent on Microsoft Windows

Figure 5.2 MySQL Enterprise Monitor: Installing Agent on Windows: Agent Authentication

6. If you chose Host and database, specify the information about the MySQL server that you want to monitor. Enter the IP address or host name of the host you want to monitor, and the port. 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. The optional Monitor Group determines the custom group that this MySQL instance will be listed under, as defined in the Service Manager.

Note
A single Agent can monitor multiple MySQL instances, which can be defined later.

Figure 5.3 MySQL Enterprise Monitor: Installing Agent on Windows: Monitored Database Information

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

Click **Next** to define the optional General and Limited users for the monitored MySQL instance.

7. Optionally, define the less privileged users for your monitored MySQL instance. Checking **Auto-Create Less Privileged Users** will automatically create the General and Limited users for you, using the username/password information that you provide. Unchecking this option requires you to manually create the users yourself, before proceeding with the Agent installation. For more information, see Section 5.2, “Creating MySQL User Accounts for the Monitor Agent”.

Figure 5.4 MySQL Enterprise Monitor: Installing Agent on Windows: Additional Agent Users

8. You receive a Configuration Report containing some of the information that you have entered during the installation. Check the information provided in the report. If the information is correct, click **Next** to continue. If you see a problem, use **Back** to go back to the configuration screen and change the information.

9. You have a final opportunity to change the installation parameters. Click **Next** to start the installation process.

10. Once the agent is installed, you get a confirmation message. Click **Next** to finalize the installation.

11. You can start the MySQL Enterprise Monitor Agent automatically now that the installation is complete. To start the agent now, leave the checkbox selected. To start the agent separately, uncheck the checkbox. Click **Finish** to exit the installation.

Once the Monitor Agent is installed, you must start it. For information on how to start and stop the Agent, see Section 5.7.1, “Starting/Stopping the Agent on Windows”.

**5.6 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 `mysqlmonitoragent-version-installer` application.
Installing the Agent on Mac OS X

- To use the installer GUI, locate the unpacked application in the Finder and double-click its icon.
- To use the text-based installer, navigate to the Contents/MacOS subdirectory inside the .app directory, and issue the command `installbuilder.sh --mode text`.

The following section describes the installation process on OS X, referring to the installation GUI.

1. First, select the language for the MySQL Enterprise Monitor Agent installation. Click OK to continue installation.
2. Click Next to start the installation process.
3. Select the installation directory. The default installation directory is /Applications/mysql/enterprise/agent. Select the installation directory, or type the new directory location.

   **Note**
   The Installer will not overwrite a current Agent installation, and instead prompts for you to either delete the existing Agent installation, or rename the installation directory.

   Also select the method that the agent uses to communicate with the MySQL server, either a TCP/IP (network) connection, or a Socket (local) connection. Choose the connection method, and click Next.

   **Note**
   The monitor agent always associates “localhost” with the TCP/IP address 127.0.0.1, not the MySQL socket. This is in contrast to the MySQL Command Line Tool, which connects via the MySQL socket by default on Unix, if the hostname “localhost” is specified.

   If the MySQL server to monitor was started with the `--skip-networking` option, then you cannot connect to it via TCP/IP, as the server does not listen for TCP/IP connections. In this case, configure the monitor agent to use the MySQL socket. During installation, select “socket” rather than “TCP/IP” and then specify the MySQL socket name. Or, after installation, reconfigure this using the `agent.sh` configuration script. For further information, refer to Section D.2, “MySQL Enterprise Monitor Agent Reference”.

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

4. Choose one or more operations for this agent to perform:
Figure 5.5 MySQL Enterprise Monitor: Agent Monitoring Options

- **Host only**: Install the Agent, but not configure a MySQL instance to monitor at this time.
- **Host and database**: Install the Agent, and also configure the initial MySQL instance to monitor.

Regardless of your choice, the Agent will continue to look for new MySQL instances to monitor on the host. The only difference is whether or not to immediately configure and monitor the first MySQL instance.

5. You may optionally install the Agent as a service. This generates a `launchd.plist` file (by default, it is stored as `/Library/LaunchDaemons/mysql.agent.plist`, and launchd has the benefit of automatically starting the Agent after the host is restarted. Using a service is recommended.
6. The Agent Installer will now install the files on your system. The following steps involve the configuration of the Agent.

7. Enter the details of the MySQL Enterprise Service Manager to use with this Agent. You can retrieve the username from the `configuration_report.txt` file generated when you installed MySQL Enterprise Service Manager, but the encrypted password is not viewable. Enter the required information and then click **Next** to continue.

**Note**

This is the user that connects the Agent to the Service Manager. It is defined by the Service Manager, and Agent users can be modified under Settings, Manage Users. Their role is defined as "agent".
8. If you chose "Host and database", specify the information about the initial MySQL instance that you want to monitor. The configuration information you enter depends on the connection method selected in the previous screen.

- If you chose TCP/IP as the connection method, enter the IP address or host name of the host to monitor, and the port, Admin user name and password to connect to the MySQL server. 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.
If you chose Socket as the connection method, enter the full path name to the Unix socket created by your MySQL server, and the user name and password to authenticate with the server. Typical values include `/tmp/mysql.sock` and `/var/mysql/mysql.sock`. 
Click **Next** to continue the installation.

9. Optionally, define the less privileged users for your monitored MySQL instance. Checking **Auto-Create Less Privileged Users** will automatically create the General and Limited users for you, using the username/password information that you provide. Unchecking this option requires you to manually create the users yourself, before proceeding with the Agent installation. For more information, see Section 5.2, “Creating MySQL User Accounts for the Monitor Agent”.
10. You receive a Configuration Report containing some of the information that you entered during the installation. Check the information provided in the report. If you see a problem, use Back to go back to the configuration screen and change the information. If the information is correct, click Next to continue.

11. You are now told how to start the Agent, which is by issuing a command similar to:

Once the Monitor Agent is installed, it needs to be started. For information on how to start and stop the Agent, see Section 5.7.2, “Starting/Stopping the Agent on Mac OS X”.

12. The final step links to the README. Press Finish to complete the installation, and close the Agent Installer.

Note
On Mac OS X, MySQL Enterprise Monitor will load the system SSL libraries instead of the bundled OpenSSL, which relates to using framework calls (e.g., launchd) to run the agent.

Note
If MySQL Enterprise Monitor Agent is installed as the root user, directories and files that the Agent writes to will be owned as the mysql user in the mysql group, which includes logs/, spool/, and etc/agentManaged. And the Agent will be started with the mysql user.

5.7 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 is not available, and MySQL Enterprise Service Manager provides a warning if an agent and the MySQL server that it monitors is unavailable.
5.7.1 Starting/Stopping the Agent on Windows

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

**Note**

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

**Warning**

To report its findings, the agent needs to be able to connect to the Monitor UI through the port specified during installation. The default value for this port is **18443**; ensure that this port is not blocked. If you need help troubleshooting the agent installation see, Section 5.12, “Troubleshooting the Agent”.

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

```
shell> sc start MySQLEnterpriseMonitorAgent
```

or:

```
shell> net start MySQLEnterpriseMonitorAgent
```

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

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

You can 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:
5.7.2 Starting/Stopping the Agent on Mac OS X

Using launchd

The preferred method is to use `launchd` to load the Agent as a service. After selecting "Install as a service" during the installation process, you may load or unload the Agent service using the following commands.

To start (load) the Agent:

```shell
sudo launchctl load /Library/LaunchDaemons/mysql.agent.plist
```

To stop (unload) the Agent:

```shell
sudo launchctl unload /Library/LaunchDaemons/mysql.agent.plist
```

Using init

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

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

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

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

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

```shell
kill -TERM PID
```

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

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

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

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

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

For further information on troubleshooting the agent, see Section 5.12, "Troubleshooting the Agent".

Installation creates the directory `/Applications/mysql/enterprise/agent`, and the `logs` directory is located immediately below the `agent` directory.
To see all the command-line options available when running the monitor agent, navigate to the 
/Applications/mysql/enterprise/agent/etc/init.d directory and execute mysql-monitor-agent help, which displays the usage message:

```
Usage: ./mysql-monitor-agent {start|stop|restart|status}
```

**Warning**
To report its findings, the agent connects to the Monitor UI through the port specified during installation. The default value for this port is 18443; ensure that this port is not blocked. To troubleshoot the agent installation, see Section 5.12, “Troubleshooting the Agent”.

### 5.7.3 Starting/Stopping the Agent on Unix

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

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

For a non-root installation the command would be:

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

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

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

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

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

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

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

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

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

For further information on troubleshooting the agent, see Section 5.12, “Troubleshooting the Agent”.

Installation creates the directory `/opt/mysql/enterprise/agent`, with the `logs` directory is located immediately below the `agent` directory.

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

```
Usage: ./mysql-monitor-agent {start|stop|restart|status}
```

**Warning**
To report its findings, the agent connects to the Monitor UI through the port specified during installation. The default value for this port is 18443; ensure that this port is not blocked. To troubleshoot the agent installation, see Section 5.12, “Troubleshooting the Agent”.
5.7.4 sql_mode

On startup, the agent sets
sql_mode=STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION,NO_AUTO_CREATE_USER
on the monitored MySQL instance. If sql_mode=ONLY_FULL_GROUP_BY, agent
queries can fail. The local agent of the MySQL Enterprise Service Manager also sets
sql_mode=STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION on the repository.

5.8 Monitoring Multiple MySQL Servers

You can monitor multiple MySQL servers (either on the same machine, or remotely across different
machines) using a single Agent.

Make sure that the MySQL instance that you want to monitor has a suitable user to use for connecting
to the host. For more information, see Section 5.2, “Creating MySQL User Accounts for the Monitor
Agent”.

Typically, an Agent will scan a host and report unmonitored MySQL instances to the MySQL Enterprise
Monitor User Interface. For more information about how this works, see Section 1.2, “MySQL
Enterprise Monitor Agent”. For information about how to change the status of a MySQL instance from
unmonitored to monitored, see Section 14.4, “MySQL Instances Dashboard”.

Note
In MySQL Enterprise Monitor 2.x, a single Agent was designed to monitor a
single MySQL instance. As of MySQL Enterprise Monitor 3.0.0, an Agent can
monitor multiple MySQL instances, and unmonitored MySQL instances are now
reported to MySQL Enterprise Service Manager using the MySQL Process
Discovery Advisor.

5.9 Configuring an Agent to Monitor a Remote MySQL Server

Typically, the Agent runs on the same machine as the MySQL servers that it is monitoring. To monitor
MySQL servers running on remote hosts, 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 Chapter 5, Monitor Agent Installation. Follow the directions given there, being
careful to either select “host-only” and add remote MySQL instances later, or 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).

Ensure that the Agent has the appropriate rights to log in to the MySQL server from a host other than
localhost and that the port used by the MySQL server, typically 3306 must be open for remote
access. For more information about the database credentials required by agents see, Section 5.2,
“Creating MySQL User Accounts for the Monitor Agent”.

The Agent also needs to be able to log in to the MySQL Enterprise Service Manager, typically using
port 18443, so ensure that the appropriate port is open.

Remote Monitoring Limitations

- Remote monitoring does not provide operating system level data, such as CPU, file, and network
  utilization information.
- Monitoring multiple MySQL instances with a single agent potentially means having a single point
  of failure. This is especially true for remote monitoring, as it might lose a connection, which means
  a black period, whereas a local Agent will continue monitoring and provides information upon
  reconnection.
Monitoring Amazon RDS

This section describes how to monitor a MySQL instance in a cloud environment, such as the Amazon Relational Database Service (Amazon RDS).

Important

It is recommended that you use MySQL 5.6, or later, on RDS. It is also possible to use MySQL 5.5, but you must disable backup and replicas before attempting to monitor it using MySQL Enterprise Service Manager. After the agent has connected, you can enable backup and replicas again.

Remote monitoring is used when monitoring on a cloud. You can use any MySQL Enterprise Monitor Agent to monitor MySQL instances remotely, including the bundled Agent that is automatically installed and started with the MySQL Enterprise Service Manager.

When configuring a MySQL instance to monitor from the MySQL Instances Dashboard, do the following:

- Do not configure MySQL Enterprise Monitor to auto-create the less privileged Limited and General accounts, and instead use the Admin account for all monitoring. This is set in the Connection Settings tab when adding or editing a MySQL instance to be monitored. The Auto-Create Less Privileged Users setting defaults to Yes, ensure it is set to No.

- Also under Connection Settings is the Instance Address parameter. Set this to your endpoint, which is the entry point for your MySQL Server web service.

- Change the inventory table schema for MySQL Enterprise Monitor Agent from "mysql" to an existing, alternative schema. This is set in the Advanced Settings tab when adding (or editing) a MySQL instance to be monitored. The Inventory Table Schema setting defaults to mysql, which is typically not accessible to the Agent user in a cloud (or shared) environment. Change it to a schema you created.

Your MySQL instance is displayed on the MySQL Instances Dashboard.

5.10 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 agent-mgmt-hostname value specified in the etc/bootstrap.properties configuration file. For more information about the contents and location of the configuration file, see Section D.2.1, “MySQL Enterprise Monitor Agent Configurable Options”. Stop the agent and change the hostname value as shown in the following:

```
agent-mgmt-hostname = https://agent_name:password@localhost:18443/
```

Replace the agent_name and password with suitable values. Likewise replace port 18443 if you are not running the Monitor UI 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:
HTTP Connection Timeout

shell> ssh -L 18443:Monitor_UI_Host:18443 -l user_name -N Monitor_UI_Host

When prompted, enter the password for user_name.

If you are not running the MySQL Enterprise Service Manager on port 18443, substitute the appropriate port number. Likewise, replace Monitor_UI_Host with the correct value. user_name represents a valid operating system user on the machine that hosts the MySQL Enterprise Service Manager.

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

- Under Windows see, Section 5.7.1, “Starting/Stopping the Agent on Windows”.
- Under Unix see, Section 5.7.3, “Starting/Stopping the Agent on Unix”.
- Under Mac OS X see, Section 5.7.2, “Starting/Stopping the Agent on Mac OS X”.

5.11 HTTP Connection Timeout

The HTTP connection between agent and Service Manager has a default timeout of 250 seconds for an attempted connection and 300 seconds for an established connection. It is possible to override these values in bootstrap.properties by adding the following parameters:

1. http-connect-timeout-ms=N: Where N is the number of milliseconds to wait before timing-out a HTTP connection attempt.
2. http-socket-timeout-ms=N: Where N is the number of milliseconds to wait before timing-out a HTTP socket read or write.

If set to zero (0), no timeout is defined. Negative values are not supported.

5.12 Troubleshooting the Agent

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

- Windows: Section 5.7.1, “Starting/Stopping the Agent on Windows”
- Unix: Section 5.7.3, “Starting/Stopping the Agent on Unix”
- Mac OS X: Section 5.7.2, “Starting/Stopping the Agent on Mac OS X”

Some additional tips are:

- To diagnose any issues with the agent, start by looking at the Logs link under the Settings tab, as described in Section 15.3, “Logs”. This page consolidates troubleshooting information across all the MySQL Enterprise Monitor components.
- To run on start-up, the agent requires correct login credentials for the monitored MySQL server. Log in to the monitored MySQL server and check the agent's credentials. Compare the values of the Host, and User fields in the mysql.user table with the values shown in the etc/agentManaged/mysqlConnection<id>/bean/json file. The passwords are encrypted so they can not be manually managed here, but the password can be altered from the MySQL Instances page in the MySQL Enterprise Monitor User Interface, or by using the agent connection tool (agent.sh) from the command line.
- 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 D.2.2, “Agent Log Files”.
- 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 D.2.2, “Agent Log Files”.
If no HTTP authentication dialog box appears, and you are unable to connect at all, then the host name or port might be specified incorrectly. Confirm the values you entered against those described as the Application hostname and port: in the configuration_report.txt file. Failure to connect could also indicate that the port is blocked on the machine hosting the MySQL Enterprise Service Manager.

- To check if a blocked port is the problem, temporarily bring down your firewall. If the agent is then able to connect, open up the port specified during installation and restart the agent. If necessary you can monitor outside the firewall using an SSH tunnel. For more information, see Section 5.10, “Monitoring Outside the Firewall with an SSH Tunnel”.

- 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.

- If the agent and the MySQL server it is monitoring are running on different machines, ensure that the correct host is specified for the agent account. The correct port, typically 3306, must also be open for remote login. For more information about remote monitoring see, Section 5.9, “Configuring an Agent to Monitor a Remote MySQL Server”.

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

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

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

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

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

### 5.13 Agent Backlog

The agent backlog is a caching mechanism which stores monitoring data in the event the agent cannot communicate with the MySQL Enterprise Service Manager. The backlog can store 20MB of monitored data, 10MB in active RAM, and 10MB on the local filesystem.

- Monitoring one MySQL instance: the agent backlog can store up to 30 minutes of monitored data before the backlog cache is filled and data dropped.

- Monitoring 10 MySQL instances: the agent backlog can store up to 3 minutes of monitored data before the backlog cache is filled and data dropped.

The agent can lose contact with the MySQL Enterprise Service Manager in the event of a Service Manager restart, network connectivity issues and so on.

Under certain circumstances, if the server loses power for example, the backlog can be corrupted and be unreadable by the agent. When the agent restarts, if the backlog files are corrupted, the monitoring agent backs up the corrupted files to a new folder named backlog.xxx. Where xxx is the UNIX
timestamp, marking the time the backup was created. The agent creates a new backlog file, and logs the error.
Chapter 6 Post-installation Considerations

Table of Contents

6.1 General Considerations ........................................................................................................ 61
6.2 Installing SSL Certificates ............................................................................................... 62
6.3 Changing an SSH Host Key ............................................................................................... 65

Depending upon your use of MySQL Enterprise Monitor, you might perform some or all of these tasks after installation.

6.1 General Considerations

This section describes some of the general tasks which may be required after installation or upgrade.

New users: Guide to completing your configuration

• Create additional users by going to Settings, Manage Users.

• Open Event Handling from the Configuration menu, and complete the SMTP configuration. Perform the test to make sure an email arrives with success. Optionally, also set up the SNMP configuration.

• On the MySQL Instances page:
  • Create new groups for your MySQL instances, such as "Production", "Development", and "QA", and then move or copy existing MySQL instances into the appropriate groups. This enables you to define Event Handling and Advisor scheduling policies on a per-group basis.
  • Create new monitoring connections either by processing the "Unmonitored" MySQL instances that were discovered on hosts with an Agent installed (via the MySQL Process Discovery Advisor), or by manually specifying connection parameters for each MySQL instance.
  • Choose Event Handling from the Configuration menu, and define handlers to route notifications to the appropriate recipients.
  • Optionally customize Advisor thresholds or exceptions on the Advisor page from the Configuration menu.
  • If you are monitoring MySQL 5.6.14+ instances, open the Query Analyzer page to view the rich SQL performance tuning data that is available by default.
    If you are monitoring earlier MySQL versions, then download a Query Analyzer plugin so you can see this performance data.
  • If you intend to monitor a large number of MySQL Instances or Hosts, stop now and re-configure the system to scale up accordingly. For performance tuning information and tips, see Chapter 9, Performance Tuning MySQL Enterprise Monitor.

Existing users: Guide to completing your upgrade

• Read Chapter 2, What’s new in MySQL Enterprise Monitor 3.0?

• On the MySQL Instances page:
  • Create new groups for your MySQL instances, such as "Production", "Development", and "QA", and then move or copy existing MySQL instances into the appropriate groups. This enables you to define Event Handling and Advisor scheduling policies on a per group basis.
Installing SSL Certificates

• Create new monitoring connections either by processing the "Unmonitored" MySQL instances that were discovered on hosts with an Agent installed (via the MySQL Process Discovery Advisor), or by manually specifying connection parameters for each MySQL instance that you want to monitor.

• Choose Event Handling from the Configuration menu, and define handlers to route notifications to the appropriate recipients. Notifications are now handled via group-based policies.

• Upgrade all of your 2.x Agents to the new 3.x version. If you are using an existing Proxy/Aggregator installation, see Chapter 11, Proxy and Aggregator Installation for information.

• Optionally customize Advisor thresholds or exceptions on the Advisor page from the Configuration menu.

• If you are monitoring MySQL 5.6.14+ instances, open the Query Analyzer page to view the rich SQL performance tuning data that is available by default.

If you are monitoring earlier MySQL versions, then download a Query Analyzer plugin so you can see this performance data.

• If you intend to monitor a large number of MySQL Instances or Hosts, stop now and re-configure the system to scale up accordingly. For performance tuning information and tips, see Chapter 9, Performance Tuning MySQL Enterprise Monitor.

6.2 Installing SSL Certificates

Important

The self-signed certificates delivered with your MySQL Enterprise Monitor installation are set to expire after 365 days. Every upgrade is delivered with new certificates set to expire 365 days after the day the upgrade package was built. In the unlikely event you are running a version of MySQL Enterprise Service Manager using the default certificates for more than a year, you must generate new certificates. If you do not generate new certificates, the SSL connection between MySQL Enterprise Service Manager and the repository fails. This section describes how to generate those certificates.

These instructions guide you through the process of installing SSL certificates for your MySQL Enterprise Monitor installation. The $INSTALL_ROOT represents the root path of your installation, which defaults to:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Default Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor\</td>
</tr>
<tr>
<td>Linux / Solaris</td>
<td>/opt/mysql/enterprise/monitor/</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/monitor/</td>
</tr>
</tbody>
</table>

Generating SSL Key and Certificate

To use SSL, you must generate a certificate and private key. These can be verified and signed through a third-party authority, such as Thawte or Entrust, or generated locally and self-signed. The recommended tool for locally-generated SSL key and certificates is the OpenSSL Toolkit. The OpenSSL libraries are delivered by default with UNIX, Linux and Mac OS X platforms, but must be obtained separately for Microsoft Windows from http://slproweb.com/products/Win32OpenSSL.html. The Windows installation also requires the Visual C++ 2008 Redistributables libraries.
For security reasons, we recommend you install the latest, compatible version of the OpenSSL Toolkit.

If you intend to use a Certificate Authority to verify your organisation's identity and sign your certificate, you must generate a private key, which is used to create a Certificate Signing Request (CSR), and send the CSR file to the Certificate Authority.

To generate the RSA private key, run the following command:

```
openssl genrsa -out insertName.key 2048
```

This generates a 2048-bit, RSA private key.

To generate the Certificate Signing Request (CSR), run the following command:

```
openssl req -new -nodes -key insertName.key -out insertName.csr
```

This command prompts for input. Complete the fields as required.

The **CN** field must correspond to the hostname. It is recommended that you use the fully-qualified server name, rather than `localhost`.

If you intend to use a self-signed certificate, you can generate the key and certificate with a single command:

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365
```

This command generates a 2048-bit RSA key, `key.pem`, and a certificate, `cert.pem`, which is valid for 365 days.

### MySQL Enterprise Service Manager

To install an SSL certificate for the MySQL Enterprise Service Manager:

Save the certificate and private key, both in PEM format, in the following location:

```
$INSTALL_ROOT/apache-tomcat/conf/ssl/tomcat.cert.pem
$INSTALL_ROOT/apache-tomcat/conf/ssl/tomcat.key.pem
```

Next, restart the service manager. For more information about stopping and starting the service manager, see the instructions for Unix/Mac OS X and Microsoft Windows.

If you are using a chained certificate implementation, you must add the following attribute to the `Connector` element of Tomcat's `server.xml`:

```
SSLCertificateChainFile="/opt/mysql/enterprise/monitor/apache-tomcat/conf/ssl/tomcat.int.pem"
```

### SSL for the Repository

For information on SSL and MySQL Server, see Creating SSL and RSA Certificates and Keys.

### MySQL Enterprise Monitor Agent

To configure SSL-related options for the Agent, the following values may be placed in `INSTALL_ROOT/etc/bootstrap.properties`:

```
```
### Table 6.2 SSL Configuration Options For The Agent’s `bootstrap.properties`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ssl-verify-hostnames</code></td>
<td>true or false</td>
<td>Verify that the hostname of the service manager that the Agent is connected to matches what is in the SSL certificate. Default is false, as we are only using SSL for confidentiality.</td>
<td></td>
</tr>
<tr>
<td><code>ssl-allow-self-signed-certs</code></td>
<td>true or false</td>
<td>If set to <code>true</code> self-signed certificates are permitted. If set to <code>false</code>, self-signed certificates are not permitted. Default value is <code>true</code>.</td>
<td></td>
</tr>
<tr>
<td><code>ssl-verify-host-certs</code></td>
<td>true or false</td>
<td>Default false, but to support self-signed certificates, a commercial certificate, or if the CA certificate has been imported into a keystore, then set to true.</td>
<td>3.0.20</td>
</tr>
<tr>
<td><code>ssl-ca-keystore-path</code></td>
<td>String</td>
<td>Path to keystore with CA cert(s), if <code>ssl-allow-self-signed-certs</code> is true. This path must be defined as a URL. For example: file:///Applications/mysql/enterprise/agent/etc/mykeystore</td>
<td></td>
</tr>
<tr>
<td><code>ssl-ca-keystore-password</code></td>
<td>String</td>
<td>Password for the CA keystore, if <code>ssl-allow-self-signed-certs</code> is true.</td>
<td></td>
</tr>
</tbody>
</table>

An example `bootstrap.properties` SSL certification section:

```properties
ssl-verify-hostname=false
ssl-allow-self-signed-certs=true
ssl-ca-keystore-path=file:///Applications/mysql/enterprise/agent/etc/mykeystore
ssl-ca-keystore-password=password123
```

To import a CA certificate in PEM format to a new keystore on the Agent, execute the following:

```
$INSTALL_ROOT/java/bin/keytool -import -file /path/to/ca/ca.pem -alias CA_ALIAS -keystore $INSTALL_ROOT/etc/cacerts
```

The tool responds with the certificate details. For example:

```
Enter keystore password:  (the keystore will require at least a 6 character password)
Re-enter new password:

Owner: CN=servername.com, O=MySQL AB, L=Uppsala, C=SE
Issuer: O=MySQL AB, L=Uppsala, ST=Uppsala, C=SE
Serial number: 100002
Certificate fingerprints:
Signature algorithm name: MD5withRSA
Version: 1
Trust this certificate? [no]: (type yes + enter)
Certificate was added to keystore
```

You must edit the `ssl-ca-*` configuration values in `bootstrap.properties` accordingly, to use the path to the keystore and password.
LDAP SSL Configuration

SSL configuration for LDAP is configured at the MySQL Enterprise Service Manager Java VM level. That is, it is configured in the keystore of the Java VM bundled with your MySQL Enterprise Monitor installation.

Important

The JVM shipped with MySQL Enterprise Service Manager does not support the AES256 cipher. This can prevent you using LDAP servers which implement that cipher.

To connect to LDAP servers which implement the AES256 cipher, you must download and install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8 package. This package is available from: Java Cryptography Extension.

The steps described in this section assume your LDAP server is correctly configured and you have a root CA certificate which was used to generate the LDAP server's certificate.

To enable SSL for LDAP and MySQL Enterprise Service Manager, you must do the following:

1. Convert the LDAP server's root CA certificate from PEM to DER format, if necessary. If the CA certificate is already in DER format, continue to the next step.

   openssl x509 -in cacert.pem -inform PEM -out ~/cacert.der -outform DER

2. Import the CA certificate, in DER format, into the MySQL Enterprise Service Manager Java keystore. Run the following command from the bin directory of your MySQL Enterprise Service Manager's Java installation:

   keytool -import -trustcacerts -alias ldapssl -file ~/cacert.der -keystore lib/security/cacerts

3. Restart MySQL Enterprise Service Manager with the following command:

   mysql/enterprise/monitor/mysqlmonitorctl.sh restart

6.3 Changing an SSH Host Key

The SSH Host key is used to distinguish monitored hosts, there should not be duplicate SSH keys. A key can be duplicated if a server is cloned. This section describes how to change the SSH host key for a particular host, eliminating the events and alarms generated when duplicate hosts are detected.

The following steps must be performed:

• Generate a new SSH key for the monitored host.

• Edit the monitoring agent's configuration.

• Edit the hostid in the MySQL Enterprise Service Manager repository.

On UNIX, Linux and Mac OS platforms, use the ssh-keygen utility. On Microsoft Windows platforms, there are several tools, but this example uses puttygen.

To generate a new SSH key for the monitored host, do the following:

1. On the monitored host, generate an SSH key. For example:

   $ ssh-keygen -t rsa -N '' -f /etc/ssh/ssh_host_key
Changing an SSH Host Key

If using puttygen, click Generate and follow the instructions on-screen.

Note

The key can be generated using RSA (SSH1 or SSH2), DSA, or ECDSA. All are supported by MySQL Enterprise Monitor.

2. Retrieve the key fingerprint.

The fingerprint is an alphanumeric string similar to the following:

```
```

On UNIX-based platforms, retrieve this value with the following command:

```
$ ssh-keygen -l -f /path/to/key/filename.pub
```

On Windows platforms, using puttygen, this value is in the Key Fingerprint field.

3. Stop the monitoring agent.

4. Open the monitoring agent’s bootstrap.properties configuration file, and add, or edit, the following value:

   ```
   agent-host-id=ssh:{New SSH Fingerprint}
   ```

   For example, using the fingerprint listed above:

   ```
   ```

5. On the MySQL Enterprise Service Manager machine, edit the hostid value in the repository:

   ```
   mysql> UPDATE mysql.inventory SET VALUE = 'ssh:{New SSH Fingerprint}' WHERE name = 'hostId';
   ```

6. Restart the monitoring agent.
Chapter 7 Upgrading, Re-Installing or Changing Your Installation

Table of Contents

7.1 General considerations when upgrading MySQL Enterprise Monitor ........................................... 67
7.2 Restoring from Backup .................................................................................................................. 69
7.3 Guide for Upgrading to MySQL Enterprise Monitor 3.0 ............................................................ 69
7.4 Upgrading an Existing 3.0.x Installation ....................................................................................... 72

This chapter describes how to upgrade from MySQL Enterprise Monitor 2.3 to 3.0 and how to upgrade an existing 3.0 installation.

7.1 General considerations when upgrading MySQL Enterprise Monitor

This section describes how to perform an update for the MySQL Enterprise Service Manager or the MySQL Enterprise Monitor Agent.

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. You must perform a fresh installation instead.

You cannot update 2.2 to 3.0. Users of 2.2 must upgrade to 2.3 before proceeding with the upgrade process.

You also cannot use the upgrader if you installed the MySQL server for the MySQL Enterprise Service Manager instance outside the MySQL Enterprise Service Manager installation directory.

Customizations to setenv.sh are lost, as this file is replaced and optimized for MySQL Enterprise Monitor 3.0 during an upgrade.

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

Important

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

Warning

The Upgrade installer for MySQL Enterprise Service Manager overwrites any changes made to the my.cnf in your MySQL Enterprise Service Manager installation. Backup the existing my.cnf file before starting the upgrade installer.

SSL Considerations

The Upgrade installer for MySQL Enterprise Service Manager overwrites any changes made to the CA certificate files within your MySQL Enterprise Service Manager installation. This includes cacerts
for the bundled JRE, and myKeystore for the bundled Tomcat web server. These files are replaced with the bundled CA certificate files, so if custom changes exist, such as an LDAP server with a self-signed certificate, you must manually re-import the custom changes. These files are backed up during the upgrade, and the locations are logged within the configuration_report.txt log file.

The following example imports one root CA from the backed up file, and uses the default password for keystores in Java which is "changeit". The alias names and path locations may also be different.

```bash
$ pwd
/opt/mysql/enterprise/monitor/java/bin
$ ./keytool -importkeystore \
   -srckeystore /opt/mysql/enterprise/monitor/backup/java/lib/security/cacerts \
   -destkeystore /opt/mysql/enterprise/monitor/java/lib/security/cacerts \
   -srcstorepass changeit \
   -deststorepass changeit \
   -srcalias myorigrootca \
   -destalias myrootca
```

The following are the options:

- **-srckeystore**: the location of the source keystore (the backed up keystore), which contains the certificate you want to import.

- **-destkeystore**: the location of the destination keystore. The certificate is imported to this location.

- **-srcstorepass**: password of the source keystore. The default password is changeit. If you changed the password, enter it here.

- **-deststorepass**: password of the destination keystore. The default password is changeit. If you changed the password, enter it here.

- **-srcalias**: the alias of the certificate in the source keystore. If you have multiple certificates in the keystore, run the following command for a list of all certificates and their aliases in the source keystore:

  ```bash
  keytool -list -v -keystore /Applications/mysql/enterprise/monitor/backup/java/lib/security/cacerts
  ```

  If source alias is not provided, all entries in the source keystore are imported and stored in the destination keystore with their original aliases.

- **-destalias**: the alias to use in the new keystore.

  **Note**
  The command is the same if you are using mykeystore, except for the paths, which must be changed to reflect the different location.

### Running the Update Installer

The name of the update file varies, but it shows the target operating system, the version the update applies to, and the component name. (There are separate update installers for the Service Manager and the Agent.) For example, a file named `mysqlenterprisemanager-3.0.0-windows-update-installer.exe` updates MySQL Enterprise Service Manager on Windows to version 3.0.0.

You 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.

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.
You can run an unattended upgrade, the same way you run an unattended install. To see all the options you can specify during the upgrade process, run the update installer with the `--help` option.

### 7.2 Restoring from Backup

This section describes how to restore an installation from a backup.

If you chose to back up your current installation, a directory named `backup` is created in the current installation directory. This directory contains copies of the directory or directories that are replaced during the update. In cases where only specific files are replaced, the `backup` directory may contain only these files. To undo the update, stop both the MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent, delete the files and directories in the installation directory, except for the `backup` directory. Copy the contents of the `backup` directory to the installation directory. Then restart the services.

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 can also choose not to back up the repository.

### 7.3 Guide for Upgrading to MySQL Enterprise Monitor 3.0

The purpose of the guide is to help you safely upgrade your production MySQL Enterprise Monitor from 2.3 to 3.0 with minimal loss of active monitoring during the upgrade process. If you are performing a complete or first-time installation, see Chapter 4, *Service Manager Installation* instead.

**Note**

The MEM 3.0 update installers upgrade from the recent versions of 2.3 or 3.0 to the latest 3.0 release. You can also use an update installer to reconfigure the same version, i.e. run the 3.0.0 update installer on 3.0.0 itself to specify a different port for the UI or change an SSL setting. If your existing Monitor setup is older than 2.3, you must either upgrade to 2.3 and then upgrade to 3.0 or perform a clean installation of 3.0.

**Quick summary**

1. It is recommended to start by installing a "test" of a 3.0 Service Manager and 3.0 Agents side-by-side with your 2.3 environment.

2. When comfortable with 3.0, you can then either:
   - Phase-out your 2.3 installation, (perhaps retaining it for historical information), and phase-in your "test" 3.0 installation as the primary Monitoring tool.
   - After testing the 3.0 installation, uninstall it and migrate your 2.3 MySQL Enterprise Service Manager to 3.0.

Running a test installation of the 3.0 Service Manager enables you to learn, configure and test the new system without disturbing your production 2.3 Monitor.

**Important**

MySQL Enterprise Monitor 3.0 has significant differences to version 2.3, with an entirely different inventory, instrument data, Query Analysis, Advisor configuration, and notification models. For this reason, when upgrading from 2.3 to 3.0, history data for events, graphs, Query Analysis, and configuration data of Advisor schedules are not migrated.

With this in mind, the following are suggested upgrade paths, which allow you to maintain your monitoring coverage with minimal loss of history.
Using a Bundled or Remote repository

When you install the 3.0 test Service Manager, you are prompted on whether or not to use the bundled MySQL server to manage the Monitor’s repository. If your existing 2.3 Monitor does use the bundled MySQL server, then you should choose the same option for your 3.0 test system.

However, if your 2.3 setup uses a separate MySQL instance that you manage, you should set up the 3.0 test install the same way by configuring another MySQL database server to use as the 3.0 test repository. Do not use the same repository as the 2.3 instance, as when the Service Manager starts, it migrates this data, and removes the tables within the MySQL Enterprise Service Manager schema. You should do this now before proceeding.

---

**Note**

Multiple MySQL Enterprise Service Manager installations cannot share the same repository, so do not attempt to share a single non-bundled repository for both a 2.3 and 3.0 installation.

**Note**

Moving the Monitor’s repository onto its own host enables the system to scale to monitor significantly more Instances.

Installing the 3.0 Service Manager in a test environment

Once it is installed, launch the Service Manager and complete the first-time setup. After a brief warm-up period, MySQL Enterprise Monitor 3.0 beginse monitoring the host and repository. Next you should follow the steps in the Chapter 6, Post-installation Considerations section, set up SMTP, user accounts and privileges, email notification groups and, depending on the size of your environment, groups of MySQL Instances.

---

**Note**

**Warm-up Period:** If a 2.3 to 3.0 Service Manager upgrade is performed while 2.3 Agents are still active, Agents and/or Instances appear to be unavailable. This triggers Events and email notifications. The Events auto-close once the warm-up period is over.

Install one Agent per monitored Host

Now that the Service Manager is fully configured, the final installation task is to install a single 3.0 Agent on each physical host that you want to include in your 3.0 test. New in 3.0, for each Agent you install, you can configure it to monitor its host (only) or optionally also configure it to monitor a MySQL Instance at install time. Whichever you choose, the Agent continuously detects and reports to the UI any unmonitored Instances it discovers whether they were present when you installed or start up in future. If you have multiple MySQL Instances running on a single host, use a single 3.0 Agent to monitor them.

Deploying 3.0 in production

When you are finished testing and are ready to deploy 3.0 as your production monitoring system, you have a couple of choices depending on how large your installation is, and whether you intend to keep running the 2.3 Monitor once 3.0 is deployed.

**Method #1: Switch 3.0 Test” to 3.0 Production**

If you have already installed, customized and tuned your 3.0 test installation, you may want to simply convert it into your production monitoring system by upgrading any remaining 2.3 Agents to 3.0 and re-directing them to the 3.0 Service Manager. See the “Upgrading Agents to 3.0” section below for how
Method #2: Shutdown 3.0 test, and upgrade 2.3 to 3.0

At this point, the 3.0 test was successful and you want to shut it down and then convert your existing 2.3 Monitor to version 3.0.

Important

The 3.0 update installer migrates application data like SMTP settings, user and notification information, Group names, Instance names and notes, etc; however, the new Event and Graph functionality are so different (and improved) from 2.3 that 3.0 cannot display 2.3 historical Graph and Event data in the UI. If you want to retain access to that historical data until it is completely purged, follow the instructions above to "Switch to "3.0 test" to "3.0 production" and leave the 2.3 Monitor running.

Upgrading from 2.3.x to 3.0.0 overwrites MySQL Enterprise Monitor's my.cnf/my.ini configuration file. The only MySQL options that are migrated from the 2.3.x MySQL configuration file are port, datadir, socket, ssl-ca, ssl-cert, ssl-key, and innodb_log_file_size.

If your 2.3 Monitor is communicating with a significant number of Agents and Instances, we advise suspending monitoring temporarily by: (1) shutting down 2.3 Agents (2) updating the Service Manager to 3.0 (3) updating each 2.3 Agent to 3.0 and watching them go live one-by-one. See the Section 7.3, “Guide for Upgrading to MySQL Enterprise Monitor 3.0” section for important information about this procedure.

After the Service Manager and Agents are upgraded to 3.0, you can uninstall the 3.0 test installation. Make sure any Instances or Hosts you were monitoring as part of the test are now pointing to your upgraded Service Manager, or uninstall those 3.0 test Agents.

Upgrading Agents to 3.0

The 3.0 Monitor runs most efficiently and effectively if all the Agents communicating with it are shutdown normally, upgraded to 3.0 and then restarted. It is strongly recommended to upgrade all Agents to the latest version. If you have a very large number of monitored Hosts or Instances, it is always best to restart them one at a time or in small batches after they are upgraded to 3.0.

Note

* The 3.0 Service Manager is designed to communicate with 2.3 Agents in a limited manner to facilitate minimal downtime during the upgrade process. There are a few issues to be aware of if one or more 2.3 agents are live and talking to a 3.0 Service Manager

Multiple Agent Accounts: Connections from the Agent to a monitored MySQL Instance are done using whatever account has the minimum permissions level required. For more information, see Section 5.2, “Creating MySQL User Accounts for the Monitor Agent”.

SSL is now required as follows:

- HTTP requests to the 3.0 UI now redirect to a secure port, and may generate an "untrusted site" warning in your browser. See SSL Considerations.

- 3.0 Agents always use a secure connection when communicating with the Service Manager. For backwards compatibility, SSL is still optional (but recommended) for 2.3 Agents connecting to the 3.0 Service Manager.
• Although SSL is optional with 2.3, for security purposes we recommend that you enable SSL for your 2.3 Agents.

The upgrade installer checks for the MEM 2.3 Agent configuration file (`mysql-monitor-agent.ini`) to verify that the directory you point to for the upgrade is an Agent installation directory.

Clone the 2.3 Monitor and upgrade it as part of your test, the steps are:

On a new system:

1. Install with a full installer for 3.0
2. Don’t start it up after the installation (just say “no” when it asks to launch the app at the end)
3. Copy the mysql data directory from 2.3, and overwrite the data directory you just installed
4. Start up the Service Manager

It migrates all user data and you start off with SMTP, SNMP, and so on, configured.

### Upgrading Agents doing multi-instance monitoring

If you are using a 2.3 Agent to monitor multiple MySQL instances, then special considerations are required when upgrading these Agents to 3.0.

**Note**

After you have tested your 3.0 installation, you may want to switch production monitoring to this system instead of going back and physically upgrading your original 2.3 system. If this is at all likely, we suggest you provision this new 3.0 Monitor host with resources (CPU, RAM, and so on) at least equivalent to the one that is currently running your 2.3 system.

When multiple 2.3 Agents are installed on a single host, first migrate one of the 2.3 Agents to 3.0 (as described in the previous documentation), and then migrate the connection details from each of the other 2.3 Agents using the following command:

```
shell> ./bin/agent.sh --migrate-agent=/path/to/some/agent/install/etc/instances
```

Now, each of the monitored MySQL instances uses the upgraded 3.0 Agent. Alternatively, you may rely on the auto-discovery method of the 3.0 Agent to discover the additional MySQL instances.

### Migrating custom rules/graphs

These are automatically migrated after MySQL Enterprise Monitor 3.0 is started.

### Proxy and Aggregator Notes

As of MySQL Enterprise Monitor 3.0.14, the MySQL Enterprise Monitor Proxy and Aggregator are delivered separately from the MySQL Enterprise Monitor Agent. To upgrade from a previous version, you must uninstall your existing MySQL Enterprise Monitor Agent and MySQL Enterprise Monitor Proxy and Aggregator installation and install the new version. For more information on the MySQL Enterprise Monitor Proxy and Aggregator, see Chapter 11, *Proxy and Aggregator Installation*

### 7.4 Upgrading an Existing 3.0.x Installation

This section describes how to upgrade an existing installation of MySQL Enterprise Monitor 3.0.x.

Each release of MySQL Enterprise Monitor contains a full installation package for a clean installation, and an upgrade installation package to upgrade existing installations to the latest version. Each
installer, full or update, with the exception of FreeBSD, also contains a compatible JVM. If you are installing on FreeBSD, you must download and install a compatible JVM.

**Upgrading the MySQL Enterprise Service Manager**

The MySQL Enterprise Service Manager upgrade installer performs the following tasks:

- Stops the MySQL Enterprise Service Manager service.
- Stops the Tomcat service.
- Stops the MySQL Enterprise Monitor MySQL repository.
- Checks to ensure the services have stopped.
- Backs up the existing installation and repository.
  
  The backup enables you to rollback to the preceding version if a problem occurred with the upgrade.
- Updates the installation with the new MySQL Enterprise Service Manager version and installs the bundled JVM.

**Upgrading the MySQL Enterprise Monitor Agent**

The MySQL Enterprise Monitor Agent upgrade installer performs the following tasks:

- Stops the running agent.
- Backs up the existing installation.
  
  The backup enables you to rollback to the preceding version if a problem occurred with the upgrade.
- Updates the installation with the new MySQL Enterprise Monitor Agent version and installs the bundled JVM.
Chapter 8 Unattended Installation Reference

Table of Contents

8.1 Unattended Installation ................................................................. 75
8.1.1 Performing an Unattended Installation ...................................... 75
8.1.2 MySQL Enterprise Service Manager Options ............................... 76
8.1.3 MySQL Enterprise Monitor Agent Options ................................ 82

8.1 Unattended Installation

This section explains how to automate the install and upgrade procedures for the MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent components, to perform those operations across one or multiple machines without any user interaction.

To perform an unattended installation, specify the installation mode as **unattended** by using the **mode** command line option. In this mode, you specify all the installation parameters, such as the installation directory, and user, password, and network options, through command-line options. For convenient scripting, you can save these options in a text file and run the installer using the **optionfile** option.

Before performing an unattended installation, familiarize yourself with the options by doing at least one interactive MySQL Enterprise Monitor install. Read the regular installation instructions, since some tasks still remain after an unattended installation: you must configure the MySQL Enterprise settings, and start up all the services/daemons.

8.1.1 Performing an Unattended Installation

The basic process for performing an unattended installation is the same for both the MySQL Enterprise Monitor Agent and MySQL Enterprise Service Manager installers, with the only difference being the options supported by each installer. For information on the options for MySQL Enterprise Service Manager, see Section 8.1.2, “MySQL Enterprise Service Manager Options”. For information on the options for MySQL Enterprise Monitor Agent, see Section 8.1.3, “MySQL Enterprise Monitor Agent Options”.

There are two methods for installation: either specify the option on the command line, or use an options file containing the relevant options and their values.

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

```
shell> mysqlmonitoragent-version-linux-x86-64bit-installer.bin
--installdir /data0/mysql/agent
--mysqlport 3306
--mysqluser root --mysqlpassword foo --agent_autocreate
--checkmysqlhost yes --managerhost localhost --managerport 48080 --agentuser AGENTUSER
--agentpassword PASSWORD --mode unattended --mysql-identity-source default
```

For unattended installation using an option file, create a text file that contains the definition for the installation. The following example uses a sample configuration file named `options.server.txt`:

```
done/installdir/opt/mysql/enterprise/install.debugtrace.monitor.log
--installdir /opt/mysql/enterprise/install.debugtrace.monitor.log
--driverclass com.mysql.jdbc.Driver
--url jdbc:mysql://localhost:3306/sakila
--remoteuser admin
--remotepassword password
mode=unattended
```

For unattended installation using an option file, create a text file that contains the definition for the installation. The following example uses a sample configuration file named `options.server.txt`:

```
done/installdir/opt/mysql/enterprise/install.debugtrace.monitor.log
--installdir /opt/mysql/enterprise/install.debugtrace.monitor.log
--driverclass com.mysql.jdbc.Driver
--url jdbc:mysql://localhost:3306/sakila
--remoteuser admin
--remotepassword password
mode=unattended
```

For unattended installation using an option file, create a text file that contains the definition for the installation. The following example uses a sample configuration file named `options.server.txt`:

```
done/installdir/opt/mysql/enterprise/install.debugtrace.monitor.log
--installdir /opt/mysql/enterprise/install.debugtrace.monitor.log
--driverclass com.mysql.jdbc.Driver
--url jdbc:mysql://localhost:3306/sakila
--remoteuser admin
--remotepassword password
mode=unattended
```
This file identifies a directory and file name for a log file, sets the mode to unattended, and uses the installdir option to specify an installation directory.

Note

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

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

Check the options in your option file closely before installation; problems during unattended installation do not produce any error messages.

Put the monitor installer file and the options file in the same directory.

The following examples show how to start the unattended installation from the command line.

On Windows within a command shell:

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

On Unix, use a command-line of the form:

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

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

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

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

As a minimum for the MySQL Enterprise Monitor Agent installation, 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.

### 8.1.2 MySQL Enterprise Service Manager Options

The following options let you customize the installation process for MySQL Enterprise Service Manager. The MySQL Enterprise Service Manager supports using a bundled MySQL server, or a separate MySQL server provided by the user. To use your own MySQL server, the server must be installed and running before installation. For more information, see Section 3.2.3, "MySQL Enterprise Monitor Repository".

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--adminpassword</td>
<td>Password for the database repository</td>
<td></td>
</tr>
<tr>
<td>--adminuser</td>
<td>Username for the database repository</td>
<td></td>
</tr>
<tr>
<td>--backupdir</td>
<td>Backup directory path.</td>
<td></td>
</tr>
<tr>
<td>--createDataBackup</td>
<td>Backup stored data. Upgrade process only.</td>
<td></td>
</tr>
</tbody>
</table>
## MySQL Enterprise Service Manager Options

### Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--dbhost</td>
<td>Hostname or IP address of the MySQL server</td>
<td></td>
</tr>
<tr>
<td>--dbname</td>
<td>Name of the repository database.</td>
<td></td>
</tr>
<tr>
<td>--dbport</td>
<td>TCP/IP port for the MySQL server</td>
<td></td>
</tr>
<tr>
<td>--debuglevel</td>
<td>Set the debug information level</td>
<td></td>
</tr>
<tr>
<td>--debugtrace</td>
<td>File for a debug trace of the installation</td>
<td></td>
</tr>
<tr>
<td>--forceRestart</td>
<td>Upgrade only. Restarts the services after the upgrade process completes.</td>
<td></td>
</tr>
<tr>
<td>--help</td>
<td>Display the list of valid options</td>
<td></td>
</tr>
<tr>
<td>--installdir</td>
<td>Installation directory</td>
<td></td>
</tr>
<tr>
<td>--installer-language</td>
<td>Language selection</td>
<td></td>
</tr>
<tr>
<td>--mode</td>
<td>Installation mode</td>
<td></td>
</tr>
<tr>
<td>--mysql_installation_type</td>
<td>MySQL server to be used by the MySQL Enterprise Monitor</td>
<td></td>
</tr>
<tr>
<td>--mysql_ssl</td>
<td>Use SSL when connecting to the database</td>
<td></td>
</tr>
<tr>
<td>--optionfile</td>
<td>Installation option file</td>
<td></td>
</tr>
<tr>
<td>--system_size</td>
<td>Defines Tomcat and MySQL repository configuration based on installation size.</td>
<td></td>
</tr>
<tr>
<td>--tomcatport</td>
<td>Server port for the Tomcat component</td>
<td></td>
</tr>
<tr>
<td>--tomcatshutdownport</td>
<td>Shutdown TCP/IP port for the Tomcat component</td>
<td>3.0.3</td>
</tr>
<tr>
<td>--tomcatsslport</td>
<td>SSL TCP/IP port for the Tomcat component</td>
<td></td>
</tr>
<tr>
<td>--unattendedmodeui</td>
<td>Unattended mode user interface</td>
<td></td>
</tr>
<tr>
<td>--version</td>
<td>Display the product information</td>
<td></td>
</tr>
</tbody>
</table>

### Command-Line Format

- **--help**
  
  Display the list of valid installer options.

- **--version**
  
  Display product and version information.

- **--backupdir**
  
  Upgrade only. The backup directory.

- **--createDataBackup**
  
  Upgrade only. The backup directory.

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--createDataBackup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>boolean</td>
</tr>
<tr>
<td>Default</td>
<td>1</td>
</tr>
<tr>
<td>Valid Values</td>
<td>0 (Do not create data backup)</td>
</tr>
</tbody>
</table>
MySQL Enterprise Service Manager Options

| 1 | (Create data backup) |

Upgrade only. Specifies whether the upgrade process should create a backup of the existing data. If `--backupdir` is not defined, a Backup directory is created in the root of the installation directory.

- **--optionfile**

**Command-Line Format**

```
--optionfile
```

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

- **--mode**

**Command-Line Format**

```
--mode
```

Permitted Values (Linux)

<table>
<thead>
<tr>
<th>Type</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>gtk</td>
</tr>
<tr>
<td>Valid Values</td>
<td>gtk (GTK (X Windows))</td>
</tr>
<tr>
<td></td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

Permitted Values (OS X)

<table>
<thead>
<tr>
<th>Type</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>osx</td>
</tr>
<tr>
<td>Valid Values</td>
<td>osx (Mac OS X (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

Permitted Values (Unix)

<table>
<thead>
<tr>
<th>Type</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>xwindow</td>
</tr>
<tr>
<td>Valid Values</td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

Permitted Values (Windows)

<table>
<thead>
<tr>
<th>Type</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>win32</td>
</tr>
<tr>
<td>Valid Values</td>
<td>win32 (Windows (native))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

The installation mode to use for this installation.

- **--debugtrace**

**Command-Line Format**

```
--debugtrace
```

Permitted Values

<table>
<thead>
<tr>
<th>Type</th>
<th>string</th>
</tr>
</thead>
</table>

The filename to use for a debug trace of the installation.

- **--debuglevel**

**Command-Line Format**

```
--debuglevel
```

Permitted Values

<table>
<thead>
<tr>
<th>Type</th>
<th>numeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>78</td>
</tr>
</tbody>
</table>
MySQL Enterprise Service Manager Options

| Min Value | 0 |
| Max Value | 4 |

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

- **--installer-language**

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

  The installer language.

- **--installdir**

  **Command-Line Format** | `--installdir`  
  **Permitted Values (OS X)** | **Type** | string  
  | **Default** | /Applications/mysql/enterprise/monitor/  
  **Permitted Values (Unix)** | **Type** | string  
  | **Default** | /opt/mysql/enterprise/monitor/  
  **Permitted Values (Windows)** | **Type** | string  
  | **Default** | C:\Program Files\MySQL\Enterprise\Monitor  

  The installation directory for MySQL Enterprise Service Manager, or the previous installation directory when performing an update. Installation only. It is not possible to change the installation directory in an upgrade.

- **--system-size**

  **Command-Line Format** | `--system-size`  
  **Permitted Values** | **Type** | string  
  | **Default** | medium  
  | **Valid Values** | small (5 to 10 MySQL Servers monitored from a laptop or low-end server with no more than 4GB of RAM.)  
  | | medium (Up to 100 MySQL Servers monitored from a medium-sized, but shared, server with 4 to 8GB of RAM.)  
  | | large (More than 100 MySQL Servers monitored from a high-end, dedicated server, with more than 8GB RAM.)  

  Defines the installation type. This choice sets parameters which suit your installation type. Installation only. It is not possible to change the system size in an upgrade.

- **--tomcatport**

  **Command-Line Format** | `--tomcatport`  
  **Permitted Values** | **Type** | numeric  
  | **Default** | 18080  

The TCP/IP port for the MySQL Enterprise Service Manager. This port is used by MySQL Enterprise Monitor Agent and as the port for the interface to the MySQL Enterprise Monitor User Interface. Installation only. It is not possible to change the Tomcat port in an upgrade.

- `--tomcatsslport`

  Command-Line Format: `--tomcatsslport`
  
<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Permitted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>numeric</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default</td>
<td>18443</td>
</tr>
</tbody>
</table>

The TCP/IP port to use for SSL communication to the MySQL Enterprise Service Manager. Installation only. It is not possible to change the Tomcat SSL port in an upgrade.

- `--mysql-identity-source`

  Command-Line Format: `--mysql-identity-source`
  
<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Permitted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default</td>
<td>default</td>
</tr>
<tr>
<td></td>
<td>Valid</td>
<td>default (Default)</td>
</tr>
<tr>
<td></td>
<td>Values</td>
<td>host_plus_datadir (host_plus_datadir)</td>
</tr>
</tbody>
</table>

The mechanism used to generate a unique identity for the MySQL instance if one does not already exist. Passing in "default" uses either the "server_uuid" variable if present, or generates a random new one. Passing in "host_plus_datadir" uses a hash of the host identity and the path to the MySQL instance's data directory, to create a unique identity.

Note

This option is only available in unattended installation mode.

Note

`host_plus_datadir` is not allowed when the Agent is remote monitoring a MySQL instance, as MySQL Enterprise Monitor is unable to definitively compute a known-unique host identity in this case.

- `--mysql_ssl`

  Command-Line Format: `--mysql-ssl`
  
<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Permitted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Valid</td>
<td>0 (Do not use SSL when connecting to the database)</td>
</tr>
<tr>
<td></td>
<td>Values</td>
<td>1 (Use SSL when connecting to the database)</td>
</tr>
</tbody>
</table>

Use SSL when connecting to the database.

- `--adminuser`

  Command-Line Format: `--adminuser`
  
<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Permitted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default</td>
<td>service_manager</td>
</tr>
</tbody>
</table>
The user name to use for connecting to the database repository used by MySQL Enterprise Service Manager. If you install the bundled MySQL server, this user is configured in the new database. If you use an existing MySQL server, specify an existing user with rights to access the database.

**Note**
The repository user name, and encrypted password (you must remember it), are stored in the `config.properties` configuration file. To locate this file on your operating system, see Section D.1.3, “The `config.properties` file”.

- **--unattendedmodeui**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--unattendedmodeui</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
<tr>
<td></td>
<td>Default none</td>
</tr>
<tr>
<td></td>
<td>Valid Values</td>
</tr>
<tr>
<td></td>
<td>none (No dialogs)</td>
</tr>
<tr>
<td></td>
<td>minimal (Critical dialogs)</td>
</tr>
<tr>
<td></td>
<td>minimalWithDialogs (Minimal UI with dialogs)</td>
</tr>
</tbody>
</table>

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

- **--adminpassword**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--adminpassword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
</tbody>
</table>

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

- **--mysql_installation_type**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--mysql-installation-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
<tr>
<td></td>
<td>Default bundled</td>
</tr>
<tr>
<td></td>
<td>Valid Values</td>
</tr>
<tr>
<td></td>
<td>bundled (Use the bundled MySQL server)</td>
</tr>
<tr>
<td></td>
<td>existing (Use an existing (user supplied) MySQL server)</td>
</tr>
</tbody>
</table>

  Specifies whether the installer should configure MySQL Enterprise Service Manager to install the bundled MySQL server, or use a MySQL server that you have already installed to store the repository data.

- **--dbport**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--dbport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type numeric</td>
</tr>
<tr>
<td></td>
<td>Default 13306</td>
</tr>
</tbody>
</table>

  The TCP/IP port for the MySQL database used to store MySQL Enterprise Service Manager repository data. If you install the bundled MySQL server, this is the port where the new database listens for connections. If you use an existing MySQL server, specify the port used for connections by that MySQL server.
MySQL Enterprise Monitor Agent Options

- **--dbhost**

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

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

- **--dbname**

  **Command-Line Format:** `--dbname`  
  **Permitted Values:** Type: string  
  **Default:** `mem`  

  The name of the MySQL Enterprise Service Manager repository.

- **--forceRestart**

  **Command-Line Format:** `--forceRestart`  
  **Permitted Values:** Type: boolean  
  **Default:** `0`  

  Force a restart of MySQL Enterprise Service Manager services.

### 8.1.3 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. The available options are detailed in the following table.

**Table 8.2 MySQL Enterprise Monitor Agent Installer Options**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--agent-installtype</td>
<td>Installation type for the agent, which can be &quot;database&quot; or &quot;standalone&quot;.</td>
<td></td>
</tr>
<tr>
<td>--agent_autocreate</td>
<td>Create an account on the monitored MySQL server to be used by the agent.</td>
<td></td>
</tr>
<tr>
<td>--agentpassword</td>
<td>Password of the agent user for connecting to the monitored MySQL server.</td>
<td></td>
</tr>
<tr>
<td>--agentservicename</td>
<td>Service name for the Agent</td>
<td></td>
</tr>
<tr>
<td>--agentuser</td>
<td>Username of the agent for connecting to the monitored MySQL server.</td>
<td></td>
</tr>
<tr>
<td>--checkmysqlhost</td>
<td>Validate the supplied MySQL hostname</td>
<td></td>
</tr>
<tr>
<td>--createBackup</td>
<td>(Upgrade only) Create backup.</td>
<td></td>
</tr>
<tr>
<td>--debuglevel</td>
<td>Set the debug information level</td>
<td></td>
</tr>
<tr>
<td>--debugtrace</td>
<td>File for a debug trace of the installation</td>
<td></td>
</tr>
<tr>
<td>--generalpassword</td>
<td>General user password for the --generaluser</td>
<td></td>
</tr>
<tr>
<td>--generaluser</td>
<td>General user username for the monitored MySQL server</td>
<td></td>
</tr>
<tr>
<td>--help</td>
<td>Display the list of valid options</td>
<td></td>
</tr>
</tbody>
</table>
### MySQL Enterprise Monitor Agent Options

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--ignore-old-proxy-aggr</td>
<td>Whether to ignore the Proxy and Aggregator while executing an upgrade.</td>
<td>3.0.14</td>
</tr>
<tr>
<td>--installdir</td>
<td>Installation directory</td>
<td></td>
</tr>
<tr>
<td>--installer-language</td>
<td>Language selection</td>
<td></td>
</tr>
<tr>
<td>--limitedpassword</td>
<td>Limited user password for the --limiteduser</td>
<td></td>
</tr>
<tr>
<td>--limiteduser</td>
<td>Limited user username for the monitored MySQL server</td>
<td></td>
</tr>
<tr>
<td>--managerhost</td>
<td>Hostname of IP address of the MySQL Enterprise Monitor server</td>
<td></td>
</tr>
<tr>
<td>--managerport</td>
<td>TCP/IP port of the MySQL Enterprise Monitor server</td>
<td></td>
</tr>
<tr>
<td>--mode</td>
<td>Installation mode</td>
<td></td>
</tr>
<tr>
<td>--mysql-identity-source</td>
<td>MySQL instance identify definition</td>
<td></td>
</tr>
<tr>
<td>--mysqlconnectiongroup</td>
<td>Sets the group for the provided MySQL connection</td>
<td></td>
</tr>
<tr>
<td>--mysqlconnnmethod</td>
<td>Connection method to the monitored MySQL server</td>
<td></td>
</tr>
<tr>
<td>--mysqlhost</td>
<td>MySQL hostname/IP address</td>
<td></td>
</tr>
<tr>
<td>--mysqlpassword</td>
<td>MySQL password for the monitored --mysquser.</td>
<td></td>
</tr>
<tr>
<td>--mysqlport</td>
<td>TCP/IP port for the monitored MySQL server</td>
<td></td>
</tr>
<tr>
<td>--mysqlsocket</td>
<td>Unix socket/Named pipe for the monitored MySQL server</td>
<td></td>
</tr>
<tr>
<td>--mysqluser</td>
<td>MySQL Administrative username for the monitored MySQL server</td>
<td></td>
</tr>
<tr>
<td>--optionfile</td>
<td>Installation option file</td>
<td></td>
</tr>
<tr>
<td>--restartImmediately</td>
<td>(Upgrade only) Restart Agent immediately after updating all files.</td>
<td></td>
</tr>
<tr>
<td>--unattendedmodeui</td>
<td>Unattended mode user interface</td>
<td></td>
</tr>
<tr>
<td>--version</td>
<td>Display the product information</td>
<td></td>
</tr>
</tbody>
</table>

- **--agentpassword**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th><code>--agentpassword</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
</tbody>
</table>

Specify the agent password to use to communicate with the MySQL Enterprise Service Manager.

- **--createBackup**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th><code>--createBackup</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type boolean</td>
</tr>
<tr>
<td>Default</td>
<td>1</td>
</tr>
</tbody>
</table>

Whether to backup the data.

**Note**

This option is only available when upgrading the Agent, and not when performing a new Agent installation.

- **--restartImmediately**
MySQL Enterprise Monitor Agent Options

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Permitted Values (OS X)</th>
<th>Type</th>
<th>Permitted Values (Unix)</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>1</td>
<td>default</td>
<td>string</td>
<td></td>
<td>string</td>
</tr>
</tbody>
</table>

Restart Agent immediately after updating all files.

Note
This option is only available when upgrading the Agent, and not when performing a new Agent installation.

• **--agentuser**
  
  **Command-Line Format**
  ```
  --agentuser
  ```

  **Permitted Values**
  ```
  Type  string
  Default agent
  ```

  Specify the agent username to use to communicate with the MySQL Enterprise Service Manager.

• **--checkmysqlhost**
  
  **Command-Line Format**
  ```
  --checkmysqlhost
  ```

  **Permitted Values**
  ```
  Type  string
  Default yes (Check host)
  Valid Values yes (Check host), no (Do not check host)
  ```

  Validate the MySQL hostname or IP address

• **--debuglevel**
  
  **Command-Line Format**
  ```
  --debuglevel
  ```

  **Permitted Values**
  ```
  Type  numeric
  Default 2
  Min Value 0
  Max Value 4
  ```

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

• **--debugtrace**
  
  **Command-Line Format**
  ```
  --debugtrace
  ```

  **Permitted Values**
  ```
  Type  string
  ```

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

• **--installdir**
  
  **Command-Line Format**
  ```
  --installdir
  ```

  **Permitted Values**
  ```
  Type  string
  Default /Applications/mysql/enterprise/agent/
  ```
MySQL Enterprise Monitor Agent Options

<table>
<thead>
<tr>
<th>Permitted Values (Windows)</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>string</td>
<td>/opt/mysql/enterprise/agent/</td>
</tr>
</tbody>
</table>

Specify the directory into which to install the software.

- **--installer-language**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--installer-language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
<tr>
<td>Default</td>
<td>en</td>
</tr>
<tr>
<td>Valid Values</td>
<td>en (English)</td>
</tr>
<tr>
<td></td>
<td>ja (Japanese)</td>
</tr>
</tbody>
</table>

Set the language to use for the installation process.

- **--managerhost**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--managerhost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
</tbody>
</table>

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

- **--managerport**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--managerport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type numeric</td>
</tr>
<tr>
<td>Default</td>
<td>18443</td>
</tr>
</tbody>
</table>

Tomcat SSL Port

- **--mode**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values (Linux)</td>
<td>Type string</td>
</tr>
<tr>
<td>Default</td>
<td>gtk</td>
</tr>
<tr>
<td>Valid Values</td>
<td>gtk (GTK (X Windows))</td>
</tr>
<tr>
<td></td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Permitted Values (OS X)</th>
<th>Type string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>osx</td>
</tr>
<tr>
<td>Valid Values</td>
<td>osx (Mac OS X (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Permitted Values (Unix)</th>
<th>Type string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>xwindow</td>
</tr>
<tr>
<td>Valid Values</td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
</tbody>
</table>
MySQL Enterprise Monitor Agent Options

<table>
<thead>
<tr>
<th>Permitted Values (Windows)</th>
<th>Type</th>
<th>Default</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>unattended</td>
<td>win32</td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

Specify the installation mode to use for this installation. The GUI is executed by default, with the possible values including text and unattended. On Linux, the GUI options are gtk (default) and xwindow.

-  `--mysqlconnmethod`

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--mysqlconnmethod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
<tr>
<td>Default</td>
<td>tcpip</td>
</tr>
<tr>
<td>Valid Values</td>
<td>tcpip (Use TCP/IP)</td>
</tr>
</tbody>
</table>

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

-  Note

This option is only available when installing the Agent, and not when performing an Agent upgrade.

-  `--mysqlhost`

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--mysqlhost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
<tr>
<td>Default</td>
<td>127.0.0.1</td>
</tr>
</tbody>
</table>

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

-  `--mysqlpassword`

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--mysqlpassword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
</tbody>
</table>

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

-  `--mysqlport`

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--mysqlport</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type numeric</td>
</tr>
<tr>
<td>Default</td>
<td>3306</td>
</tr>
</tbody>
</table>

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

-  `--mysqlsocket`
MySQL Enterprise Monitor Agent Options

Permitted Values | Type  | string  
---|---|---

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

• **--mysqluser**

Command-Line Format | **--mysqluser**
Permitted Values | Type  | string  

The MySQL Server administrative user for the MySQL instance to be monitored. This user must already exist.

• **--agent_autocreate**

Command-Line Format | **--agent-autocreate**
Permitted Values | Type  | boolean  

Auto-create the less privileged users (**--generaluser** and **--limiteduser**) using the **--mysqluser** user. Use this option if the limited and general users do not already exist on your system.

The default value depends on the context. For new installations, it is "1", and for upgrades it is "0".

• **--generaluser**

Command-Line Format | **--generaluser**
Permitted Values | Type  | string  

The username for the general user.

• **--generalpassword**

Command-Line Format | **--generalpassword**
Permitted Values | Type  | string  

Password for the **--generaluser**.

• **--limiteduser**

Command-Line Format | **--limiteduser**
Permitted Values | Type  | string  

The username for the limited user.

• **--limitedpassword**

Command-Line Format | **--limitedpassword**
Permitted Values | Type  | string  

Password for the **--limiteduser**.

• **--optionfile <optionfile>**

Command-Line Format | **--optionfile**
Permitted Values | Type  | string  

Specify the location of an option file containing the configuration options for this installation.
### MySQL Enterprise Monitor Agent Options

- **--unattendedmodeui**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--unattendedmodeui</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
<tr>
<td></td>
<td>Default none</td>
</tr>
<tr>
<td></td>
<td>Valid Values</td>
</tr>
<tr>
<td></td>
<td>none (No dialogs)</td>
</tr>
<tr>
<td></td>
<td>minimal (Critical dialogs)</td>
</tr>
<tr>
<td></td>
<td>minimalWithDialogs</td>
</tr>
</tbody>
</table>

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

- **--use-external-glib**

<table>
<thead>
<tr>
<th>Removed</th>
<th>3.0.14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
</tbody>
</table>

Specifies to use the `glib` library already present on the installation machine, and not to install the one bundled with MySQL Enterprise Monitor Agent.

- **--version**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--version</th>
</tr>
</thead>
</table>

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

- **--agent_installtype**

<table>
<thead>
<tr>
<th>Command-Line Format</th>
<th>--agent-installtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted Values</td>
<td>Type string</td>
</tr>
<tr>
<td></td>
<td>Default database</td>
</tr>
</tbody>
</table>

Installation type for the Agent. Passing in "standalone" configures the Agent to only monitor the Host itself. Passing in "database" configures the Agent to monitor both the Host and a specific MySQL Instance.

This option is typically used when setting "--mode" to "unattended".

**Note**

Additional MySQL Instances can be added for monitoring in the future.

- **--ignore-old-proxy-aggr**

<table>
<thead>
<tr>
<th>Removed</th>
<th>3.0.14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--ignore-old-proxy-aggr</td>
</tr>
<tr>
<td>Permitted Values</td>
<td>Type boolean</td>
</tr>
<tr>
<td></td>
<td>Default 0</td>
</tr>
</tbody>
</table>

Ignores Proxy and Aggregator while running an upgrade.
MySQL Enterprise Monitor Agent Options

Note
This option is only available when upgrading the Agent, and not when performing a new Agent installation.

- **--mysqlconnectiongroup**

  Command-Line Format: `--mysqlconnectiongroup`
  Permitted Values:

<table>
<thead>
<tr>
<th>Type</th>
<th>Permitted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Optionally sets the MySQL instance group for the connection.</td>
</tr>
<tr>
<td></td>
<td>As of 3.0.5, multiple groups can be assigned in a single installation by passing in a comma-separated list of group names.</td>
</tr>
</tbody>
</table>

- **--agentservicename**

  Command-Line Format: `--agentservicename`
  Permitted Values:

<table>
<thead>
<tr>
<th>Type</th>
<th>Permitted Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>(Linux)</td>
</tr>
<tr>
<td>Default</td>
<td><code>mysql-monitor-agent</code></td>
</tr>
<tr>
<td>string</td>
<td>(OS X)</td>
</tr>
<tr>
<td>Default</td>
<td><code>mysql.monitor.agent</code></td>
</tr>
<tr>
<td>string</td>
<td>(Unix)</td>
</tr>
<tr>
<td>Default</td>
<td><code>mysql-monitor-agent</code></td>
</tr>
<tr>
<td>string</td>
<td>(Windows)</td>
</tr>
<tr>
<td>Default</td>
<td><code>MySQL Enterprise Monitor Agent</code></td>
</tr>
</tbody>
</table>

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

The default value is `mysql-monitor-agent`.

Note
This option is only available when installing the Agent, and not when performing an Agent upgrade.

- **--help**

  Command-Line Format: `--help`
  Display the list of valid options to the installer.
There are two major components of the Service Manager that require tuning, the MySQL Instance that is used for the Repository, and the Apache Tomcat application server that serves the Web UI and performs the back-end collection and analysis of data.

9.1 Tuning Memory

This section describes how to adjust the resources available to your MySQL Enterprise Service Manager installation.

Tuning Tomcat

If you experience MySQL Enterprise Service Manager performance issues, increasing the amount of RAM available to the JVM installed with Tomcat can resolve those issues. The JVM memory settings are defined by the `JAVA_OPTS` line of the `setenv` file which sets the environment variables for Tomcat.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tomcat Heap Size</td>
<td>256MB</td>
<td>768MB</td>
<td>2048MB</td>
</tr>
<tr>
<td>Tomcat MaxPermSize</td>
<td>200MB</td>
<td>512MB</td>
<td>1024MB</td>
</tr>
</tbody>
</table>

The minimum and maximum heap size are set to the same value to have all the available memory set for the Tomcat JVM's sole use from startup.

- `--JvmMs` (Windows)/`-Xms` (all other platforms): sets the minimum size of the Tomcat JVM heap.
- `--JvmMx` (Windows)/`-Xmx` (all other platforms): sets the maximum size of the Tomcat JVM heap.

This can be adjusted depending on the size of your installation, and the free memory on the host that MySQL Enterprise Service Manager is installed upon. For example, if you have installed the MySQL
Enterprise Service Manager on a well-resourced server with a 64-bit operating system, 64GB of RAM, and are monitoring more than 100 agents, increasing the heap size to 5 or 6GB may be necessary. This depends on the MySQL server load, and amount of data collected by the agents.

Important

If you change these settings, you must restart the MySQL Enterprise Service Manager.

The following are examples of medium-sized, default settings, as defined by the medium installation choice:

**UNIX, Linux, and Mac:**

```
JAVA_OPTS=-Xmx768M -Xms768M -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/opt/mysql/enterprise/monitor/apache-tomcat/temp
-XX:+UseParallelOldGC -XX:MaxPermSize=512M
```

**Microsoft Windows:**

```
set JAVA_OPTS=--JvmMs 768 --JvmMx 768 ++JvmOptions=-XX:+UseParallelOldGC
++JvmOptions=-XX:+HeapDumpOnOutOfMemoryError ++JvmOptions=-XX:HeapDumpPath=@@BITROCK_TOMCAT_ROOTDIR@@
temp
++JvmOptions=-XX:MaxPermSize=512M
```

If the MySQL Enterprise Service Manager is insufficiently resourced, the monitoring agents are also affected. If the agents are unable to communicate with the MySQL Enterprise Service Manager, their performance also degrades.

### Tuning InnoDB Memory

The MySQL Enterprise Monitor repository uses the InnoDB storage engine. The installation process sets a default value for InnoDB based on the installation size. Tuning the InnoDB Buffer Pool can have a significant impact on performance, for both interaction with the Web UI, and overall resource requirements on the host.

The configuration file for the MEM MySQL repository can be found in the following locations:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor\mysql\my.ini</td>
</tr>
<tr>
<td>Linux / Solaris</td>
<td>home/mysql/enterprise/monitor/mysql/my.cnf</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/monitor/mysql/my.cnf</td>
</tr>
</tbody>
</table>

It is possible to increase the value of the `innodb_buffer_pool_size` variable to as high as 80% of the physical memory available on the host machine. It is not recommended to raise it higher.

### 9.2 Tuning CPU

If both Apache Tomcat server and MySQL repository are installed on the same host, the best option within large scale environments is to move the MySQL Instance to its own host. This enables both processes to use up the resources of each host, and enables scaling to monitor hundreds of MySQL Instances and Hosts.

To do this, you should:

1. Stop the application Apache Tomcat server and MySQL Instance.
2. Copy the `datadir` contents to the new host (if moving to a fresh MySQL instance), or run `mysqldump` and import the dump into the new MySQL instance.
3. Modify Tomcat’s configuration to use the new MySQL Instance on the new host.

The configuration should be updated with the configuration tool (as it encrypts the password for the repository instance), this tool can be found at the following location*:

Table 9.4 MEM repository configuration tool location (default)

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td><code>C:\Program Files\MySQL\Enterprise\Monitor\bin\config.bat</code></td>
</tr>
<tr>
<td>Linux / Solaris</td>
<td><code>/opt/mysql/enterprise/monitor/bin/config.sh</code></td>
</tr>
<tr>
<td>Mac OS X</td>
<td><code>/Applications/mysql/enterprise/monitor/bin/config.sh</code></td>
</tr>
</tbody>
</table>

This Service Manager repository tool has the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--help</td>
<td>Prints this usage message</td>
</tr>
<tr>
<td>--md, --mysql-db</td>
<td>MySQL database for the Service Manager repository</td>
</tr>
<tr>
<td>--mp, --mysql-port</td>
<td>MySQL port for the Service Manager repository</td>
</tr>
<tr>
<td>--ms, --mysql-server</td>
<td>MySQL server for the Service Manager repository</td>
</tr>
<tr>
<td>--mu, --mysql-user</td>
<td>MySQL username for the Service Manager repository</td>
</tr>
<tr>
<td>-v, --version</td>
<td>Displays the version of the agent and components</td>
</tr>
</tbody>
</table>

To update the configuration, run the script in the following way:

```
shell> ./config.sh --mysql-server=[new host] --mysql-port=[new port] --mysql-user=[new user]
```

**Important**

The configuration script must be run by the same user as the MySQL Enterprise Service Manager.

You are prompted to enter the password for the new user, and the repository configuration is updated. Once finished, restart the Apache Tomcat server.

### 9.3 Tuning Apache Tomcat Threads

When monitoring with a large number of Agent processes deployed, the default number of threads that are created within the Apache Tomcat server may not be sufficient. By default, it is configured to create 150 threads to communicate with the HTTPS port.

This is configured with the `maxThreads` setting within the `server.xml` configuration file*:

Table 9.5 MEM repository configuration tool location (default)

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td><code>C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\conf\server.xml</code></td>
</tr>
<tr>
<td>Linux / Solaris</td>
<td><code>/opt/mysql/enterprise/monitor/apache-tomcat/conf/server.xml</code></td>
</tr>
<tr>
<td>Mac OS X</td>
<td><code>/Applications/mysql/enterprise/monitor/apache-tomcat/conf/server.xml</code></td>
</tr>
</tbody>
</table>

The following section should be modified:
A good baseline to test would be the number of Agents that you have checking in to the Service Manager plus 50. For example if you have 150 Agents checking in, set the `maxThreads` variable to 200.

**Note**

* We list default paths to configuration files and tools, so adjust according to where the Service Manager was installed on your system.

### 9.4 Tuning Agent Memory Requirements

The following are the recommended settings for MySQL Enterprise Monitor Agent:

- A single agent, with default settings and all advisors enabled, should monitor no more than 10 MySQL instances.

- If the agent is monitoring more than 10 MySQL instances, the agent heapsize must be increased by 64MB for every 10 additional MySQL instances.

- The `data-reporting-threads` parameter must be increased by 2 for every 15-20 MySQL instances monitored.
Chapter 10 Uninstalling the MySQL Enterprise Monitor

Table of Contents

10.1 Removing the MySQL Enterprise Monitor: Windows ........................................ 95
10.2 Removing the MySQL Enterprise Monitor: Unix ........................................ 96
10.3 Removing the MySQL Enterprise Monitor Mac OS X ................................. 98
10.4 Unattended Uninstallations ................................................................. 99

Removing the MySQL Enterprise Monitor requires removing the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent Service. In some circumstances, such as when running multiple agents on one machine, you might remove only a single monitored server rather than the entire MySQL Enterprise Monitor Agent Service.

10.1 Removing the MySQL Enterprise Monitor: Windows

Removing the MySQL Enterprise Service Manager

Remove the MySQL Enterprise Service Manager by going to the Control Panel and choosing Add or Remove Programs. Find the entry for MySQL Enterprise Monitor and remove it. During the uninstall process you are prompted to save existing data and log files. Choose this option if you plan to reinstall MySQL Enterprise Monitor.

If you are not saving existing data, you can delete the C:\Program Files\MySQL\Enterprise\Monitor directory after removing MySQL Enterprise Service Manager.

Warning

If you did not 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 deletes these files.

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

Removing MySQL Enterprise Monitor Services Only

When the MySQL Enterprise Service Manager is installed, the Tomcat/Apache and MySQL server services are started. You can remove these services without also removing your MySQL Enterprise Service Manager installation. For more information about these services, see Starting/Stopping the MySQL Enterprise Monitor Services on Windows.

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

To confirm that these services have been removed, check in the Microsoft Management Console Services window.

To reinstall these services, use the Install MySQL Enterprise Monitor Services menu option.

You can also 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: mysqlnetworkctl help. This batch file is discussed in more detail in Starting/Stopping the MySQL Enterprise Monitor Services on Windows.
Removing the Monitor Agent

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

Warning
To remove only one of the agents from a machine that is running several agents, do not remove the MySQL Enterprise Monitor Agent entry from the Add or Remove Programs menu. To remove a single agent, see Removing a Single Agent.

Removing the Monitor Agent automatically deletes its associated .log and .pid files. After removing the Monitor Agent, you might need to remove the directories, C:\Program Files\MySQL\Enterprise and C:\Program Files\MySQL\Enterprise\Agent.

Removing the Monitor Agent this way removes the default service. If you are running additional Monitor Agents, you must remove those agents manually. See the next section for instructions on doing this.

Removing a Single Agent

To remove only one of the agents from a machine that is running several agents, do not remove the MySQL Enterprise Monitor Agent entry from the Add or Remove Programs menu. To remove a single agent and leave other agents intact, follow these steps:

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

It is best to stop the agent before removing it; for instructions on stopping an agent see, Section 5.7.1, “Starting/Stopping the Agent on Windows”.

To confirm the location of the agent log files, check the ini file. For more information on this topic, see Section D.2.1, “MySQL Enterprise Monitor Agent Configurable Options”.

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

```
shell> sc delete AgentName
```

To confirm that the agent has been removed, check that there is no longer any entry for that agent in the Microsoft Management Console Services window.

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 way.

10.2 Removing the MySQL Enterprise Monitor: Unix

Removing the MySQL Enterprise Service Manager

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

Execute this file by typing:
Removing the Monitor Agent

During the uninstall process you are prompted to save existing data and log files. Choose this option if you plan to reinstall MySQL Enterprise Monitor.

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

**Warning**

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

On Red Hat Enterprise Linux 4 and Fedora Core 4, the uninstall script might not stop the Tomcat server. Do this manually if necessary. To do this, see Section 4.4, “Starting/Stopping the MySQL Enterprise Monitor Services”.

Be careful not to accidentally stop any other Java processes running on your system.

On some Unix platforms, you might have to manually delete the `uninstall` application and the installation directory after you execute the uninstall process.

### Removing the Monitor Agent

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

The `uninstall` file is located in the `/opt/mysql/enterprise/agent` directory. Execute this file by navigating to this directory and typing:

```
shell> ./uninstall
```

Removing the Monitor Agent automatically deletes its associated `.log` and `.pid` files. After uninstalling the Monitor Agent, you can remove the `/opt/mysql/enterprise/agent` directory.

Removing the Monitor Agent this way removes the default service, and all the configuration files for different instances.

### Removing a Single Agent

To remove only one of the agents from a machine that is running several agents, do **not** run the uninstall program. To remove a single agent and leave other agents intact, follow these steps:

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

It is best to stop the agent before removing it; for instructions on stopping an agent, see Section 5.7.3, “Starting/Stopping the Agent on Unix”.

To confirm the location of the agent log files, check the `ini` file. For more information on this topic, see Section D.2.1, “MySQL Enterprise Monitor Agent Configurable Options”.

To remove the agent as a daemon, remove its entry in the `init.d` directory. Also remove or archive any log or configuration files associated with this agent.
10.3 Removing the MySQL Enterprise Monitor Mac OS X

Removing the MySQL Enterprise Service Manager

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

During the uninstall process you are prompted to save existing data and log files. Choose this option if you plan to reinstall the MySQL Enterprise Monitor.

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

**Warning**

If you did not remove existing data and log files when uninstalling the MySQL Enterprise Monitor, do not remove the `/Applications/mysql/enterprise/monitor` directory; doing so deletes these files.

Removing the Monitor Agent

Prior to removing the MySQL Enterprise Monitor Agent, stop any agents by changing to the `init.d` directory and issuing the command:

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

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

Removing the Monitor Agent automatically deletes its associated `.log` and `.pid` files. After uninstalling the MySQL Enterprise Monitor Agent, you can remove the `/Applications/mysql/enterprise/agent` directory.

Removing the MySQL Enterprise Monitor Agent this way removes the default service, and all the configuration files for different instances.

Removing a Single Agent

To remove only one of the agents from a machine that is running several agents, do not run the uninstall program. To remove a single agent and leave other agents intact, follow these steps:

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

It is best to stop the agent before removing it; for instructions on stopping an agent, see Section 5.7.2, “Starting/Stopping the Agent on Mac OS X”.

To confirm the location of the agent log files, check the `.ini` file. For more information on this topic, see Section D.2.1, “MySQL Enterprise Monitor Agent Configurable Options”.

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

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 way.

## 10.4 Unattended Uninstallations

This section describes how to uninstall MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent as an unattended process. The unattended uninstallation can be run from the command line.

Both MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent have identical uninstallation options. To display those options, from the command line run the `uninstall` file in your installation directory, with the `--help` option.

The following options are available:

### Table 10.1 MySQL Enterprise Monitor Uninstaller Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--help</code></td>
<td>Displays the list of options.</td>
</tr>
<tr>
<td><code>--version</code></td>
<td>Displays the product name and version.</td>
</tr>
<tr>
<td><code>--debuglevel &lt;debuglevel&gt;</code></td>
<td>Sets the verbosity of the uninstallation log. 0 is the lowest, 4 the highest, and 2 is the default.</td>
</tr>
<tr>
<td><code>--mode &lt;mode&gt;</code></td>
<td>Sets the uninstallation mode. This varies according to the platform. For example, on Linux-based systems, you can choose a GUI-based uninstaller with <code>--mode gtk</code>, or choose a text-only, console-based uninstallation with <code>--mode text</code>.</td>
</tr>
<tr>
<td><code>--debugtrace &lt;debugtrace&gt;</code></td>
<td>Sets the path and filename of the uninstallation log file.</td>
</tr>
<tr>
<td><code>--installer-language</code></td>
<td>Sets the language of the uninstallation. Possible values are:</td>
</tr>
<tr>
<td></td>
<td><code>en</code>: English. Default value.</td>
</tr>
<tr>
<td></td>
<td><code>ja</code>: Japanese.</td>
</tr>
</tbody>
</table>

### Unattended Uninstallation

To run an unattended uninstallation process, in which no dialogs, prompts or warnings are displayed, run the following command in the installation directory of your MySQL Enterprise Service Manager or MySQL Enterprise Monitor Agent:

```sh
cShell>./uninstall --mode unattended
```
Chapter 11 Proxy and Aggregator Installation

Table of Contents

11.1 Proxy Aggregator Architecture ................................................................. 101
11.2 Prerequisites ......................................................................................... 102
11.3 Installing the Proxy and Aggregator ......................................................... 103
11.4 Graphical Installation Wizard ................................................................. 103
11.5 Text-Based Installation ......................................................................... 105
11.6 Unattended Installation ......................................................................... 106
11.7 Starting and Stopping the Proxy and Aggregator ................................. 108
11.8 Configuration Options .......................................................................... 109

This chapter describes the architecture of the various Proxy, Aggregator and Connector installations and the installation process for the Proxy and Aggregator components.

The MySQL Enterprise Monitor Aggregator requires a framework, or chassis, to handle the communications between the client application and the MySQL instance. The following frameworks are available:

- MySQL Enterprise Monitor Proxy: the Proxy functions as the communications chassis for the Aggregator and is responsible for intercepting the communications between the client application and the MySQL instance. This enables the Aggregator to collect the raw query data sent from the client application to the MySQL instance. The MySQL Enterprise Monitor Proxy and Aggregator installer can install and configure both Proxy and Aggregator, or a standalone Aggregator if one of the MySQL connectors is used as the communications chassis. The client application must be configured to communicate with the MySQL Enterprise Monitor Proxy.

- MySQL Connectors: the MySQL Connectors enable communication between the client application and the MySQL instance. If you intend to use a MySQL Connector as the communications framework for the MySQL Enterprise Monitor Aggregator, you must configure the Connector to communicate with the Aggregator. If you use a Connector with the Aggregator, you do not need to install the MySQL Enterprise Monitor Proxy.

11.1 Proxy Aggregator Architecture

This section describes the MySQL Enterprise Monitor Proxy and Aggregator architecture.

Default Architecture

The following diagram shows the MySQL Enterprise Monitor Proxy and Aggregator architecture.
Note
The MySQL Enterprise Monitor Proxy and Aggregator does not have to be installed on the same host as the monitored MySQL instance. You can install it on another host.

MySQL Enterprise Monitor Aggregator with Connector

Important
The MySQL Enterprise Monitor Aggregator is supported by the PHP Connector. The other Connectors do not require the Aggregator and can communicate directly with the MySQL Enterprise Service Manager once configured to do so. For more information on configuring the Connectors, see Chapter 12, Configuring Connectors.

11.2 Prerequisites

Important
If you are using the MySQL Enterprise Monitor 2.3 implementation of the Agent and Aggregator with a 3.0 MySQL Enterprise Service Manager, you must uninstall the 2.3 version before installing the MySQL Enterprise Monitor Proxy and Aggregator delivered with MySQL Enterprise Monitor 3.0.14.

Proxy and Aggregator Users

It is not recommended to install the MySQL Enterprise Monitor Proxy and Aggregator as root. It is recommended to create a user specifically for the Proxy and Aggregator and install the products as
Performance Schema

If you are using the MySQL Enterprise Monitor Proxy and Aggregator to collect query performance data, you must ensure the `statements_digest` consumer in `performance_schema.setup_consumers` is disabled.

11.3 Installing the Proxy and Aggregator

The following installations are possible:

- **Aggregator and Proxy**: Proxy and Aggregator are installed and configured together.
- **Aggregator**: Aggregator is installed without the Proxy. Only use this installation type if you intend to use the Aggregator with MySQL Connector/PHP.

The installer has the following filename convention:

```
mysqlmonitoraggregator-version_number-platform-architecture-installer.extension
```

where:

- `version_number` is the version number of the product.
- `platform` is the intended operating system for the installer.
- `architecture` specified whether the installer is for 32- or 64-bit platforms. If no architecture is present, the installer is 32-bit.

The installers support the following installation types:

- Graphical Installation Wizard
- Text mode
- Unattended mode

11.4 Graphical Installation Wizard

This section describes how to install the MySQL Enterprise Monitor Proxy and Aggregator together using the MySQL Enterprise Monitor Proxy and Aggregator Installation Wizard. This process is identical across all platforms, except where explicitly stated.

This installation package installs one of the following:

- **Proxy and Aggregator**: installs both the Proxy and Aggregator.
- **Aggregator Only**: installs the Aggregator only.

To install the MySQL Enterprise Monitor Proxy and Aggregator using the Graphical Installation Wizard, do the following:
Starting the Installation

1. Run the installer as required by the operating system.
   
   The language selection dialog is displayed. Choose a language and click Next.

2. On the Welcome dialog, click Forward.
   
   The Installation Directory dialog is displayed.

3. Define an installation directory, or accept the default installation directory, and click Forward.
   
   The component selection page is displayed.

   If you choose Proxy and Aggregator, you must follow the steps in Installing the Proxy and Installing the Aggregator.

   If you choose Aggregator Only, you must follow the steps in Installing the Aggregator.

   Important

   There is no difference in the files installed. The Aggregator Only option installs all files, Proxy included, but only configures the Aggregator. The Proxy files are installed, but it is not configured or started by this installation choice. If you choose Aggregator Only and attempt to start the proxy, it will not start unless correctly configured.

4. Choose your installation type and click Forward.

Installing the Proxy

This section describes how to install the MySQL Enterprise Monitor Proxy. To install the MySQL Enterprise Monitor Proxy, do the following:

1. Enter the port number the Proxy uses to listen for incoming connections. The default port is 6446.

2. Select the communication protocol the Proxy uses to connect to the monitored MySQL instance.

   Note

   Socket is not available on Windows platforms.

   If you intend to use socket to connect to the database, select Socket and click Forward to define the path to the socket you want to use. After the socket is defined, click Forward to proceed with the installation.

   If you intend to use TCP/IP, select TCP/IP and click Forward to proceed with the installation.

   The MySQL database configuration dialog is displayed.

3. Enter the hostname or IP address and the port number of the monitored MySQL instance.

   Click Forward.

   The Aggregator configuration dialog is displayed.
Installing the Aggregator

This section describes how to install the MySQL Enterprise Monitor Aggregator. To install the MySQL Enterprise Monitor Aggregator, do the following:

1. Complete the following fields on the Aggregator configuration dialog:
   - **Aggregator Port**: the port the Aggregator listens on.
   - **Aggregator SSL Port**: the port the Aggregator listens on for SSL communication.
   - **PEM Certificate file**: the location of the PEM certificate.
   - **CA Certificate file**: the location of the CA certificate.
   
   Click **Forward** to continue.

   The MySQL Enterprise Monitor options dialog is displayed.

2. Complete the MySQL Enterprise Monitor option fields. This information is used by the Aggregator to connect to the MySQL Enterprise Service Manager.
   
   You must provide the following information:
   - **Hostname or IP address**: the address of the MySQL Enterprise Service Manager installation.
   - **Tomcat SSL Port**: the port Tomcat is listening on for SSL connections.
   - **Agent Username**: the username of the Agent. These are the connection credentials the Aggregator uses to connect to the MySQL Enterprise Service Manager.
   - **Agent Password**: the password of the Agent.
   
   Click **Forward**. The **Configuration Report** is displayed.

3. Review the data in the **Configuration Report** to ensure all configuration settings are correct.

   Click **Forward** to complete the installation.

11.5 Text-Based Installation

The steps and options of the text-based installation are identical to those described in Section 11.4, "Graphical Installation Wizard".

**Note**

There is no text-mode installation available for Microsoft Windows platforms.

To start the text-based installer, do the following:

1. Run the installer with the following option:

   ```
   --mode text
   ```

   The following example shows how to start the text-mode installation on a 64-bit Linux system:

   ```
   shell>./mysqlmonitoraggregator-3.0.14.3040-linux-x86-64bit-installer.bin --mode text
   ```

   The text installation process starts.
2. Follow the instructions onscreen. The options and values are identical to those described in Section 11.4, “Graphical Installation Wizard”.

11.6 Unattended Installation

The MySQL Enterprise Monitor Proxy and Aggregator installers enable you to perform unattended installations. This is useful for large scale installations on multiple machines. The installations can be run using all required options on a command line, or by defining the required options in a configuration file and calling that file for each installation.

Unattended Installation Options

To display the installation options available, run the installer from the command line with the following option:

```
--help
```

The following options are available:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--help</code></td>
<td>Displays the help text listing all options available for the platform on which the installer was run. On Microsoft Windows platforms, this option does not output the list of options in the console window, but in a separate help window.</td>
</tr>
<tr>
<td><code>--version</code></td>
<td>Displays the product version. On Microsoft Windows platforms, this option does not output the version details in the console window, but in a separate help window.</td>
</tr>
<tr>
<td><code>--debuglevel</code></td>
<td>Sets the verbosity of the installation log. 0 is the lowest verbosity, 4 is the highest. Default value is 2.</td>
</tr>
<tr>
<td><code>--debugtrace</code></td>
<td>Sets the path and filename of the installation log file.</td>
</tr>
<tr>
<td><code>--optionfile</code></td>
<td>Sets the path and filename of the installation options file. More information, see Unattended Installation with Options File.</td>
</tr>
<tr>
<td><code>--installer-language</code></td>
<td>Sets the language of the installation. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• <code>en</code>: English. Default value.</td>
</tr>
<tr>
<td></td>
<td>• <code>ja</code>: Japanese.</td>
</tr>
<tr>
<td><code>--mode</code></td>
<td>Sets the installation mode. This varies according to the platform. For example, on Linux-based systems, you can choose a GUI-based installer with <code>--mode gtk</code>, or choose a text-only, console-based installation with <code>--mode text</code>. The following is a list of the GUI-based installation options available:</td>
</tr>
<tr>
<td></td>
<td>• Windows: <code>Win32</code></td>
</tr>
<tr>
<td></td>
<td>• OS X: <code>osx</code></td>
</tr>
<tr>
<td></td>
<td>• Solaris: <code>xwindow</code></td>
</tr>
<tr>
<td></td>
<td>• Linux: <code>gtk</code> (Default) and <code>xwindow</code>.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--mode</td>
<td><strong>--mode</strong> can also initiate text mode and unattended installations.</td>
</tr>
<tr>
<td></td>
<td>• <strong>--mode text</strong>: starts a text-only, console-based installation process. Text-based installation are not available on Windows platforms.</td>
</tr>
<tr>
<td></td>
<td>• <strong>--mode unattended</strong>: starts an unattended installation. For more information on unattended installations, see Unattended Installation from the Command Line and Unattended Installation with Options File.</td>
</tr>
<tr>
<td>--unattendedmodeui</td>
<td>Sets the graphical elements to use, if any, in the unattended installation. The following options are available:</td>
</tr>
<tr>
<td></td>
<td>• none: no pop-ups, or progress bars are displayed. Errors are displayed, if they occur.</td>
</tr>
<tr>
<td></td>
<td>• minimal: No user interaction is required and a progress bar is displayed showing the installation progress. Errors are displayed, if they occur.</td>
</tr>
<tr>
<td></td>
<td>• minimalWithDialogs:</td>
</tr>
<tr>
<td>--installdir</td>
<td>Sets the installation directory for the product.</td>
</tr>
<tr>
<td>--use-external-glib</td>
<td>Sets the glib to use, the one delivered in the installer (0, default), or the system glib (1).</td>
</tr>
<tr>
<td>--monitorcomponent</td>
<td>Specifies which component to install. The following options are available:</td>
</tr>
<tr>
<td></td>
<td>• proxy: installs both Proxy and Aggregator. This is the default.</td>
</tr>
<tr>
<td></td>
<td>• aggregator: installs the Aggregator only.</td>
</tr>
</tbody>
</table>

**Proxy-Specific Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxyservicename</td>
<td>Sets the unique service name for the Proxy service.</td>
</tr>
<tr>
<td>--mysqlconnmethod</td>
<td>Sets the connection method used by the proxy to connect to the monitored MySQL instance. The following options are available:</td>
</tr>
<tr>
<td></td>
<td>• tcpip: default value.</td>
</tr>
<tr>
<td></td>
<td>• socket: unavailable on Microsoft Windows platforms.</td>
</tr>
<tr>
<td>--proxyport</td>
<td>Sets the port the Proxy listens on for incoming connections. Default value is 6446.</td>
</tr>
<tr>
<td>--mysqlhost</td>
<td>Sets the hostname or IP address of the monitored MySQL instance. Default value is localhost.</td>
</tr>
<tr>
<td>--mysqlport</td>
<td>Sets the port of the monitored MySQL instance. Default value is 3306.</td>
</tr>
<tr>
<td>--mysqlsocket</td>
<td>Sets the socket used by the monitored MySQL instance.</td>
</tr>
</tbody>
</table>

**Aggregator-specific Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--aggregatorservicename</td>
<td>Sets the unique name for the Aggregator service. Default value is mysql-monitor-aggregator.</td>
</tr>
</tbody>
</table>
### Option Table

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--aggregatorport</td>
<td>Sets the port the Aggregator listens on. Default value is <strong>14000</strong>.</td>
</tr>
<tr>
<td>--aggregatorsslport</td>
<td>Sets the SSL port the Aggregator listens on for secure connections. Default value is <strong>14443</strong>.</td>
</tr>
<tr>
<td>--aggregatorsslcertfile</td>
<td>Sets the location of the SSL certificate.</td>
</tr>
<tr>
<td>--aggregatorsslcafile</td>
<td>Sets the location of the SSL CA file.</td>
</tr>
<tr>
<td>--managerhost</td>
<td>Sets the hostname or IP address of the MySQL Enterprise Service Manager installation. Default value is <strong>localhost</strong>.</td>
</tr>
<tr>
<td>--managerport</td>
<td>Sets the SSL port number of the MySQL Enterprise Service Manager's Tomcat installation. Default value is <strong>18443</strong>.</td>
</tr>
<tr>
<td>--agentuser</td>
<td>Sets the agent username which the Aggregator uses to communicate with the MySQL Enterprise Service Manager. Default value is <strong>agent</strong>.</td>
</tr>
<tr>
<td>--agentpassword</td>
<td>Sets the password of the agent used by the Aggregator.</td>
</tr>
</tbody>
</table>

### Unattended Installation from the Command Line

To run the unattended installation from the command line, enter the installer name, followed by the **--mode unattended** option, followed by the options you want to define. If you do not define an option on the command line, the default value is used, if a default exists. If no default value exists, you must define that value in the configuration after the installation is complete.

The following example installs MySQL Enterprise Monitor Proxy and Aggregator on a Linux platform but changes the MySQL Enterprise Service Manager values:

```
./mysqlmonitoraggregator-3.0.14.3041-linux-x86-b4bit-installer.bin --mode unattended
--unattendedmodeui none --managerhost service.manager.com --agentuser Agent100
--agentpassword D4unKotR
```

This example changes the following:

- Instructs the installer to display no dialogs of any kind. In this mode, errors are displayed if they occur.
- Sets the MySQL Enterprise Service Manager location to **service.manager.com**. This is the location of your MySQL Enterprise Service Manager installation. The default ports were not changed.
- Sets the Agent username to Agent100 and the Agent password to D4unKotR.

### Unattended Installation with Options File

If you use an options file, you add the options you want to change to a text file as name=value pairs. Using the example shown in **Unattended Installation from the Command Line**, the text file contents are:

```bash
mode=unattended
unattendedmodeui=none
managerhost=service.manager.com
agentuser=Agent100
agentpassword=D4unKotR
```

If this file was saved as **pa-options.txt**, the installation command takes the following format:

```
./mysqlmonitoraggregator-3.0.14.3041-linux-x86-b4bit-installer.bin --optionfile pa-options.txt
```

### 11.7 Starting and Stopping the Proxy and Aggregator

This section describes how to start and stop the MySQL Enterprise Monitor Proxy and Aggregator.
On UNIX, Linux, and Mac OS X platforms, the Proxy and Aggregator processes are controlled using the scripts in the `etc/init.d` directory of your installation. On Windows platforms, you can start, stop and restart your services using the Start menu entries, or through the Services control of the Microsoft Management Console.

Important

If you install the MySQL Enterprise Monitor Proxy and Aggregator, both Proxy and Aggregator run under the name of the MySQL Enterprise Monitor Proxy, not as two distinct services. If you install the MySQL Enterprise Monitor Aggregator standalone, it is run as the MySQL Enterprise Monitor Aggregator.

- **init.d**:
  - Starting the Proxy and Aggregator: run `./mysql-monitor-proxy start`.
  - Stopping the Proxy and Aggregator: run `./mysql-monitor-proxy stop`.
  - Starting the Aggregator: run `./mysql-monitor-aggregator start`.
    
    If you have installed both Proxy and Aggregator, do not run this command. The Aggregator is started by the Proxy-specific commands.
  - Stopping the Aggregator: run `./mysql-monitor-aggregator stop`.
    
    If you have installed both Proxy and Aggregator, do not run this command. The Aggregator is stopped by the Proxy-specific commands.
  - Status: run either script with the `status` option to see the status of the service.
    
    If you installed both Proxy and Aggregator, the status returns information on the Proxy only. In this installation type, if the Proxy is running, the Aggregator is running also. For more information, check the `mysql-monitor-proxy.log`.
  - Restarting: run either script, depending on your installation type, with the `restart` option to restart the services.

### 11.8 Configuration Options

It is possible to run the Proxy, or Aggregator, or both, with specific options, using the following files installed in the `bin` directory of your installation:

- `mysql-monitor-aggregator`
- `mysql-monitor-proxy`

Note

On Windows platforms, these files are executables and have the `.exe` extension. On Linux, UNIX and Mac platforms, they are shell scripts.

To view the options available, run either file with the `--help` option.

The help output is broken down into the following sections:

- Help Options: lists the various help output options.
- Application Options: lists the application options.
- `aggr-module`: lists the Aggregator-specific options. Displayed only for the `--help-all` option.
• proxy-module: lists the Proxy-specific options. Displayed only for the --help-all option.

The mysql-monitor-aggregator help displays the application and Proxy module help, only. The mysql-monitor-proxy help displays application, aggregator and proxy output.

### Table 11.2 Proxy and Aggregator Help Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Lists the basic help options.</td>
</tr>
<tr>
<td>--help-all</td>
<td>Lists all available help options.</td>
</tr>
<tr>
<td>--help-aggr</td>
<td>Lists the Aggregator-specific help options.</td>
</tr>
<tr>
<td>--help-proxy</td>
<td>Lists the Proxy-specific help options. This option is only available on the mysql-monitor-proxy file.</td>
</tr>
</tbody>
</table>

### Table 11.3 Application Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-V, --version</td>
<td>Shows the version of the Proxy or Aggregator, depending on which file it is run with.</td>
</tr>
<tr>
<td>--defaults-file=&lt;file&gt;</td>
<td>Defines a configuration file to use. Similarly to running an unattended installation with an options file, this enables you to define all configuration changes as name-value pairs (without the -- prefix for each option) and call the file as needed.</td>
</tr>
<tr>
<td>--verbose-shutdown</td>
<td>Configures the application to always log the exit code on shutdown.</td>
</tr>
<tr>
<td>--daemon</td>
<td>Configures the application to run in daemon mode.</td>
</tr>
<tr>
<td>--user=&lt;user&gt;</td>
<td>Defines the specific user to run the Aggregator.</td>
</tr>
<tr>
<td>--basedir=&lt;absolute path&gt;</td>
<td>Defines the absolute path of the base directory which is prefixed to all relative paths in the configuration. If you define a relative path, an error is returned.</td>
</tr>
<tr>
<td>--pid-file=&lt;file&gt;</td>
<td>Defines the name of the PID file to use in the event the application is started in daemon mode.</td>
</tr>
<tr>
<td>--plugin-dir=&lt;path&gt;</td>
<td>Defines the path to the plugins.</td>
</tr>
<tr>
<td>--plugins-name=&lt;name&gt;</td>
<td>Defines the names of the plugins to load. On the command line, you can specify this value multiple time. In the configuration file, the option is entered once, followed by a comma-separated list of the required plugins.</td>
</tr>
<tr>
<td>--log-level=&lt;string&gt;</td>
<td>Defines the logging level. Possible values are critical (default value), error, warning, info, message, and debug.</td>
</tr>
<tr>
<td>--log-file=&lt;filename&gt;</td>
<td>Defines the name of the logfile.</td>
</tr>
<tr>
<td>--log-use-syslog</td>
<td>Configures the application to send all messages to the syslog. UNIX/Linux only.</td>
</tr>
<tr>
<td>--log-backtrace-on-crash</td>
<td>Configures the application to invoke the debugger in the event of a crash.</td>
</tr>
<tr>
<td>--keepalive</td>
<td>Configures the application to attempt a restart in the event of a crash. Not available on Microsoft Windows. When running as a service, the Proxy automatically restarts.</td>
</tr>
<tr>
<td>--max-open-files</td>
<td>Configures the maximum number of open files.</td>
</tr>
<tr>
<td>--event-threads</td>
<td>Configures the number of event-handling threads. Default value is 1.</td>
</tr>
<tr>
<td>--lua-path=&lt;path&gt;</td>
<td>Sets the LUA_PATH.</td>
</tr>
<tr>
<td>--lua-cpath=&lt;path&gt;</td>
<td>Sets the LUA_CPATH.</td>
</tr>
</tbody>
</table>
### Table 11.4 aggr-module Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--aggr-address=<a href="">host:port</a></td>
<td>Defines the address and listening port of the Aggregator. The default port value is 14000.</td>
</tr>
<tr>
<td>--aggr-lua-script=&lt;filename&gt;</td>
<td>Defines the path to the LUA script.</td>
</tr>
<tr>
<td>--aggr-mem-url=&lt;url&gt;</td>
<td>Defines the URL to the MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td>--aggr-mem-user=&lt;string&gt;</td>
<td>Defines the Agent username to use for communication with the MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td>--aggr-mem-password=&lt;string&gt;</td>
<td>Defines the Agent password to use for communication with the MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td>--aggr-ssl-address=<a href="">host:port</a></td>
<td>Defines the address and listening port of the Aggregator for SSL connections to the Aggregator.</td>
</tr>
<tr>
<td>--aggr-ssl-cert-file=&lt;filename&gt;</td>
<td>Defines the PEM server certificate for the Aggregator.</td>
</tr>
<tr>
<td>--aggr-ssl-cs-file=&lt;filename&gt;</td>
<td>Defines the CA certificate for the Aggregator.</td>
</tr>
<tr>
<td>--aggr-ssl-ciphers=&lt;string&gt;</td>
<td>Defines the supported ciphers.</td>
</tr>
<tr>
<td>--aggr-test-mode</td>
<td>Start the Aggregator in test mode. This mode ignores the flush interval setting and aggregates queries until instructed to return the aggregated data by a HTTP REST interface. It returns a JSON result set of all the normalized queries and their aggregated data.</td>
</tr>
<tr>
<td>--aggr-flush-interval=&lt;seconds&gt;</td>
<td>Defines the interval, in seconds, at which the query data is flushed to the MySQL Enterprise Service Manager. The default value is 60 seconds.</td>
</tr>
<tr>
<td>--aggr-max-request-body-size=&lt;bytes&gt;</td>
<td>Defines the maximum size of an HTTP request body. The default size is 1MB.</td>
</tr>
</tbody>
</table>

### Table 11.5 proxy-module Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-P, --proxy-address=<a href="">host:port</a></td>
<td>The address and listening port of the Proxy. Default port is 4040.</td>
</tr>
<tr>
<td>-r, --proxy-read-only-backend-addresses</td>
<td>The address and listening port of the remote, slave server. This is not set by default.</td>
</tr>
<tr>
<td>-b, --proxy-backend-addresses=<a href="">host:port</a></td>
<td>The host name (or IP address) and port of the MySQL server to connect to. You can specify multiple backend servers by supplying multiple options. Clients are connected to each backend server in round-robin fashion.</td>
</tr>
<tr>
<td>--proxy-skip-profiling</td>
<td>Disable query profiling (statistics time tracking). The default is for tracking to be enabled.</td>
</tr>
<tr>
<td>-s file_name, --proxy-lua-script=&lt;file&gt;</td>
<td>The Lua script file to be loaded. The script file is not loaded and parsed until a connection is made. Also note that the specified Lua script is</td>
</tr>
</tbody>
</table>
### Configuration Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--no-proxy</td>
<td>Disables the Proxy module. By default, the Proxy is enabled.</td>
</tr>
<tr>
<td>--proxy-pool-no-change-user</td>
<td>Disable use of the MySQL protocol CHANGE_USER command when reusing a connection from the pool of connections specified by the proxy-backend-addresses list.</td>
</tr>
<tr>
<td>--proxy-connect-timeout</td>
<td>Defines the Proxy's connection timeout in seconds. Default value is 2.</td>
</tr>
<tr>
<td>--proxy-read-timeout</td>
<td>Defines the read timeout in seconds. Default is 8 hours.</td>
</tr>
<tr>
<td>--proxy-write-timeout</td>
<td>Defines the write timeout in seconds. Default is 8 hours.</td>
</tr>
</tbody>
</table>

These options, with the exception of the `help`, `version` and `defaults-file` options, are also used, as name=value pairs, in the ini files used to configure the Proxy and Aggregator services.

The configuration files are located in the `etc` directory of your installation.

- `mysql-monitor-proxy.ini`: configures the Proxy and Aggregator. Use this file when both components are installed.
- `mysql-monitor-aggregator.ini`: configures the Aggregator. Use this file when only the Aggregator is installed.
Chapter 12 Configuring Connectors

Table of Contents

12.1 Using the MySQL Enterprise Plugin for Connector/PHP ............................................ 113
12.2 Using the MySQL Enterprise Plugin for Connector/J .................................................. 117
12.3 Using the MySQL Enterprise Plugin for Connector/Net .............................................. 121

This section describes how to configure the Connectors to pass query information to the Query Analyzer. The following configurations are described:

- **Section 12.1, “Using the MySQL Enterprise Plugin for Connector/PHP”**: describes the configuration of a PHP-based application which uses the MySQL Enterprise Plugin for Connector/PHP and MySQL Enterprise Monitor Aggregator to feed query information to the Query Analyzer.

- **Section 12.2, “Using the MySQL Enterprise Plugin for Connector/J”**: describes the configuration of the Connector/J to feed query information to the Query Analyzer.

- **Section 12.3, “Using the MySQL Enterprise Plugin for Connector/Net”**: describes the configuration of the Connector/.NET to feed query information to the Query Analyzer.

### 12.1 Using the MySQL Enterprise Plugin for Connector/PHP

The MySQL Enterprise Plugin for Connector/PHP enables you to use the Query Analyzer to monitor MySQL queries from PHP applications, such as PHP-enabled web pages. The Query Analyzer enables you to locate and analyze queries that are inefficient or slow. Tuning such queries helps to shorten load times for web pages, and improves overall system responsiveness and scalability.

The PHP query data is routed through the MySQL Enterprise Monitor Aggregator. The Aggregator receives query information from the PHP plugin, aggregates and computes statistics, and sends this data to the MySQL Enterprise Service Manager, where it is displayed by the Query Analyzer. You must have the MySQL Enterprise Monitor Aggregator enabled and running to use Query Analyzer with PHP applications.

**Figure 12.1 Plugin for PHP and Aggregator Architecture**

---

Important

The PHP Connector is the only connector which requires the MySQL Enterprise Monitor Aggregator to aggregate queries and transmit them to the MySQL.
Prerequisites

The MySQL Enterprise Plugin for Connector/PHP requires PHP 5.3.2 or above, with the MySQL native driver, **mysqlnd**, installed. This is the recommended configuration. If your PHP installation was not configured with the **mysqlnd** enabled, you must rebuild and install PHP from source using at least one of the following options:

- `--with-mysqli=mysqlnd`
- `--with-pdo-mysql=mysqlnd`
- `--with-mysql=mysqlnd`

The preceding options are supplied to the `configure` command, depending on which extension you are using (**mysql**, **mysqli** or **PDO_MYSQL**). If you use more than one extension, provide multiple options. Specifying any of the options listed rebuilds PHP with **mysqlnd** support. You also must enable the PHP JSON module.

The MySQL client application user, that makes PHP connections in your PHP code, must have **SELECT** privileges on the **mysql.inventory** table. This table contains the server UUID required to report the Query Analyzer data to the MySQL Enterprise Service Manager. Use the **GRANT** statement. For example:

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

Installation

The plugin is provided as a regular PHP module (PHP extension), and installation follows those PHP standard procedures as described on [http://php.net/install.pecl](http://php.net/install.pecl).

Download the MySQL Enterprise Plugin for Connector/PHP, then use the following step-by-step instructions to install and configure the MySQL Enterprise Plugin for Connector/PHP extension.

1. Locate your **php.ini** configuration file. If you do not know the location, you can view information about your PHP installation by creating a script containing:

```php
<?php phpinfo(); ?>
```

Place the script within a directory configured for providing PHP web pages. Now load the page in your web browser to see a list of configuration and other information about your PHP installation.

Check the output for **Loaded Configuration File**. If the value is **(none)**, refer to the **Configuration File (php.ini) Path** and create a file called **php.ini** in there. If a **Scan this dir for additional .ini files** option is listed you can also create a file using any name you like, ending in .ini, in that directory to set configuration options.

2. Identify whether or not your PHP build was built “thread safe” by checking the **Thread Safety** value in the output from the `phpinfo()` test. If your PHP build is thread safe, you need
   - **mysqlenterprise_ts.so** on Linux, Unix, and OS X,
   - **php_mysqlenterprise_ts.dll** on Microsoft Windows.
   If not, use
   - **mysqlenterprise.so** on Linux, Unix, and OS X,
   - **php_mysqlenterprise.dll** on Microsoft Windows.

3. Add an entry for the MySQL Enterprise Plugin for Connector/PHP module. The following example uses the full path:
Alternatively, add the file to the directory defined by the `extension_dir` configuration option, and specify the filename:

```
extension=mysqlenterprise.so
```

### Note

If `mysqlnd` is loaded as a shared library (`mysqlnd.so`), then it must be loaded before `mysqlenterprise.so` or errors such as "PHP Warning: PHP Startup: Unable to load dynamic library '/mysqlenterprise.so' - /mysqlenterprise.so: undefined symbol: mysqlnd_plugin_register in Unknown on line 0" will be emitted by PHP. Either:

- If `php.ini` is used to load the PHP extensions, then list it first. For example:

  ```ini
  extension=mysqlnd.so
  extension=mysqlenterprise.so
  ```

- If individual ini files are used to load the PHP extensions, then note that the ini files are loaded alphabetically, so adjust accordingly so that `mysqlnd.so` is loaded first. For example, `/etc/php.d/` might contain:

  ```ini
  mysqlnd.ini
  mysqlzz_enterprise.ini
  ```

4. Users of Debian-based systems, such as Ubuntu, are encouraged to use the `php5enmod` command to enable extensions. For example:

   ```bash
   $ php5enmod /path/to/mysqlenterprise.so
   ```

   `php5enmod` creates a symlink from the usual `conf.d` directory that points to where the real files are located in `mods-available`, and prefixes it with a priority number.

5. Restart your Web server application to reload PHP and the configured extensions.

6. Reload the `phpinfo()` page, and inspect the listing for the `mysqlenterprise` module.

### Caution

If you are using PHP on Microsoft Windows with the Apache web server (httpd) built from apache.org, note the following:

MySQL no longer supports VC6, the MySQL Enterprise Plugin for Connector/PHP for Microsoft Windows is compiled with the newer VC9 compiler. You can not use PHP as a loaded module with an Apache web server build that uses VC6. Alternative Apache builds exist that use VC9. Check your source and ensure that your binaries are compiled using VC9.

PHP binaries for Microsoft Windows from php.net have compiled `mysqlnd` support by default, since PHP 5.3.0.

### Configuration

The configuration of the MySQL Enterprise Plugin for Connector/PHP is handled through the standard PHP configuration files, either globally using `php.ini`, or by using the per-directory options, as detailed in [PHP Configuration](#). The following table shows the available configurable options.
Note
Each PHP configuration option for MySQL Enterprise Monitor is prefixed by mysqlenterprise.

Table 12.1 Connector/PHP Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregator_connect_timeout_sec</td>
<td>Timeout, in seconds, for communications with the MySQL Enterprise Monitor Aggregator.</td>
</tr>
<tr>
<td></td>
<td>• Property type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1</td>
</tr>
<tr>
<td></td>
<td>This property can be combined with the aggregator_connect_timeout_usec property.</td>
</tr>
<tr>
<td>aggregator_connect_timeout_usec</td>
<td>Timeout, in microseconds, for communications with the MySQL Enterprise Monitor Aggregator.</td>
</tr>
<tr>
<td></td>
<td>• Property type: integer</td>
</tr>
<tr>
<td></td>
<td>• Default value: 0</td>
</tr>
<tr>
<td></td>
<td>This property can be combined with the aggregator_connect_timeout_sec property.</td>
</tr>
<tr>
<td>aggregator_user</td>
<td>The Aggregator's username. See Chapter 11, Proxy and Aggregator Installation for more information.</td>
</tr>
<tr>
<td></td>
<td>• Property type: string</td>
</tr>
<tr>
<td>aggregator_password</td>
<td>The Aggregator’s password.</td>
</tr>
<tr>
<td></td>
<td>• Property type: string</td>
</tr>
<tr>
<td>aggregator_url</td>
<td>The IP address, or hostname, and port of the Aggregator installation.</td>
</tr>
<tr>
<td></td>
<td>• Property type: string</td>
</tr>
<tr>
<td></td>
<td>• Default value: tcp://127.0.0.1:14000</td>
</tr>
<tr>
<td>debug_callback</td>
<td>This property should be used only when debugging your MySQL Enterprise Monitor installation with MySQL Support personnel.</td>
</tr>
<tr>
<td></td>
<td>Defines the name of the callback function to invoke when data is sent to the Aggregator. The callback is defined in the PHP application and is a function which requires a single parameter, the array of HTTP requests made to the Aggregator.</td>
</tr>
<tr>
<td>disable_backtrace</td>
<td>Defines whether a backtrace is performed. Backtrace is useful for debugging but has a performance impact.</td>
</tr>
<tr>
<td></td>
<td>• Property type: boolean</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1, the backtrace is disabled. To enable the backtrace, set this value to 0.</td>
</tr>
<tr>
<td>log_file</td>
<td>Defines the location of a log file which logs all query information sent to the Aggregator.</td>
</tr>
</tbody>
</table>
Using the MySQL Enterprise Plugin for Connector/J

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This should only be used for debugging purposes because every request is logged, resulting in a very large log file.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>string</strong></td>
</tr>
<tr>
<td>quan_enabled</td>
<td>Defines whether query analysis is enabled.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>boolean</strong></td>
</tr>
<tr>
<td></td>
<td>• Default value: <strong>1</strong>, query analysis is enabled. To disable query analysis, set this value to <strong>0</strong>.</td>
</tr>
</tbody>
</table>

The following is an example of the Aggregator-specific section of the **php.ini**:

```ini
extension = /usr/local/apache/php/lib/php/extensions/mysqlenterprise.so
mysqlenterprise.aggregator_url = tcp://aggregator:14000
mysqlenterprise.quan_enabled = 1
mysqlenterprise.debug_callback = cta_callback
mysqlenterprise.disable_backtrace = 1
mysqlenterprise.aggregator_user = username
mysqlenterprise.aggregator_password = "password"
```

**Note**

You must restart your server after setting these properties. Verify the settings are correct by checking the output of `phpinfo()`.

### 12.2 Using the MySQL Enterprise Plugin for Connector/J

The MySQL Enterprise Plugin for Connector/J, enables query analysis for your applications without requiring any modification to the application code.

**Important**

The Connector/J does not require the MySQL Enterprise Monitor Aggregator for query aggregation.

**Figure 12.2 Connector Plugin Architecture**
Prerequisites

1. MySQL Connector/J version 5.1.12 or later.
2. JDK-1.7.0 or later.
3. MySQL Enterprise Service Manager version 3.0 or later.
4. The MySQL client application user must have SELECT privileges on the `mysql.inventory` table. This table contains the server UUID which is required to report the Query Analyzer data to the MySQL Enterprise Service Manager. Use the `GRANT` statement. For example:

```
mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
```
5. Apache Commons logging in the `CLASSPATH` of the application being analyzed. If you are not already using Commons Logging, modify the application's `CLASSPATH` as described in the following section to point to the JAR file bundled with the MySQL Enterprise Monitor product.

Installation

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

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

There is static shutdown() method on `com.mysql.etools.jdbc.StatementPerformanceCounters`, which can be used to cleanly shutdown the query analysis plugin when the application is going to be shutdown.

If the application is deployed in a J(2)EE application server, there is a ContextListener distributed with the plugin which calls this method when the application's context is shutdown (or reloaded). Application Servers which support `@WebListener` (such as JEE6 and above) do not need to do any extra configuration, but users with older Application Servers need to add the following line to their application's `web.xml` file:

```
</listener>
```

Using the MySQL Enterprise Plugin for Connector/J

This section describes how to configure the MySQL Plugin for Connector/J.

Table 12.2 MySQL Plugin for Connector/J Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>statementInterceptors</td>
<td>Enables the plugin. Set this property as follows:</td>
</tr>
<tr>
<td></td>
<td><code>statementInterceptors = com.mysql.etools.jdbc.StatementPerformanceCounters</code></td>
</tr>
</tbody>
</table>
Using the MySQL Enterprise Plugin for Connector/J

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disableSourceLocation</td>
<td>Defines whether to send stack traces with example queries to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>boolean</code></td>
</tr>
<tr>
<td></td>
<td>• Values: <code>true</code> or <code>false</code> (default).</td>
</tr>
<tr>
<td>serviceManagerUrl</td>
<td>Defines the URL of the MySQL Enterprise Service Manager. Include the full URL and port number.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>string</code></td>
</tr>
<tr>
<td></td>
<td>• Value: URL and port number of MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td>serviceManagerUser</td>
<td>Defines the Agent username to use when connecting to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>string</code></td>
</tr>
<tr>
<td>serviceManagerPassword</td>
<td>Defines the Agent password to use when connecting to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>string</code></td>
</tr>
<tr>
<td>serviceManagerConnectTimeout</td>
<td>Defines the number of seconds to wait for a connection to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>numeric</code></td>
</tr>
<tr>
<td></td>
<td>• Default value: 0</td>
</tr>
<tr>
<td>serviceManagerResponseTimeout</td>
<td>Defines the number of seconds to wait for a response from MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>numeric</code></td>
</tr>
<tr>
<td></td>
<td>• Default value: 0</td>
</tr>
<tr>
<td>mysqlServerUUID</td>
<td>If you are unable to retrieve the server's UUID, define it with this property.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>string</code></td>
</tr>
<tr>
<td></td>
<td>To retrieve the UUID, the plugin requires <code>SELECT</code> privileges on <code>mysql.inventory</code>.</td>
</tr>
</tbody>
</table>

You can also configure MySQL Enterprise Plugin for Connector/J to use SSL for all communication with MySQL Enterprise Service Manager. To enable SSL, add the following properties to your connection string:

Table 12.3 MySQL Plugin for Connector/J SSL Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>verifySslHostnames</td>
<td>If set to true, it enables verification of the host names in the SSL Server certifi as used by Firefox, and Curl, and specified by RFC 2818.</td>
</tr>
</tbody>
</table>
## Using the MySQL Enterprise Plugin for Connector/J

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Property type</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>verifySslCerts</td>
<td>Defines whether the plugin verifies the certificate presented by the server was signed by a CA in the trustCertificateKeystore.</td>
<td>boolean</td>
<td>false</td>
</tr>
<tr>
<td>trustCertificateKeystoreUrl</td>
<td>Defines the URL of the trusted root certificate KeyStore. If none is specified, the Java defaults are used.</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>trustCertificateKeystorePassword</td>
<td>Defines the password for the KeyStore.</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>trustCertificateKeystoreType</td>
<td>Defines the KeyStore type for trusted root certificates. If type is set to NULL or empty, JKS is used by default. The standard keystore types supported by the JVM are JKS and PKCS12. Your environment may have more available depending on what security products are installed and available to the JVM.</td>
<td>[type]</td>
<td></td>
</tr>
<tr>
<td>clientCertificateKeystoreUrl</td>
<td>Defines the URL of the client KeyStore. If none specified, Java defaults are used.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>clientCertificateKeystorePassword</td>
<td>Defines the password to use for the client certificate store.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>clientCertificateKeystoreType</td>
<td>Defines the KeyStore type for client certificates. If type is set to NULL or empty, JKS is used by default.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following example configures a Connector/J to communicate with the MySQL Enterprise Service Manager localhost, on port 18443, using the agent username `agent`, and password `PASSWORD`. Add the properties to your connection string on a single line:

```java
statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters
&serviceManagerUrl=https://localhost:18443/
&serviceManagerUser=agent
&serviceManagerPassword=PASSWORD
```

You must also add the application-specific properties to the JDBC URL. For example, the following fragment connects to the MySQL database test on localhost, using the user and password of `mysqltest`, while also collecting query data and sending it to the MySQL Enterprise Service Manager on localhost:

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
Connection conn = null;
...
try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
```
Using the MySQL Enterprise Plugin for Connector/Net

```java
"?user=mysqltest" +
"&password=mysqltest" +
"&statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters" +
"&serviceManagerUrl=https://localhost:18443/* +
"&serviceManagerUser=agent" +
"&serviceManagerPassword=PASSWORD"
);

// Do something with the Connection
...
} catch (SQLException ex) {
    // handle any errors
}
```

**Note**

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

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

- `logger=Log4JLogger` (for applications using Log4J)
- `logger=CommonsLogger` (for applications using Apache Commons Logging)
- `logger=Jdk14Logger` (for applications using Java 1.4 or later logging)

Additional configuration of the plugin is done in the Enterprise Service Manager's user interface, using the **Query Analyzer** tab. From here, the capture of query performance data for a given MySQL instance that the plugin is being with can be enabled or disabled. It is also possible to configure the level of detail that is captured; summary, examples (with source code locations) and **EXPLAIN** plans.

### 12.3 Using the MySQL Enterprise Plugin for Connector/Net

The MySQL Enterprise Plugin for Connector/Net enables you to use the Query Analyzer to monitor MySQL queries from any application using Connector/Net, including both standalone and web-based applications. As described in Section 23.3, “Query Analyzer User Interface”, the Query Analyzer can help you locate queries that are inefficient or slow. Tuning such queries helps to shorten load times for web pages, and improves overall system responsiveness and scalability.

Information about the queries is sent directly to the MySQL Enterprise Service Manager. Once you install the MySQL Enterprise Plugin for Connector/Net, query analysis becomes available for your applications without requiring any modification to the application code.
Prerequisites

- You must be using Connector/Net version 6.2.3 or later.

- The MySQL client application user must have `SELECT` privileges on the `mysql.inventory` table. This table contains the server UUID; it is required to report the Query Analyzer data to the MySQL Enterprise Service Manager. Use the `GRANT` statement. For example:

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

- Your application should already be using the `Mysql.data.dll` and have been built with the library requirement.

- If you are using the released builds of Connector/Net, you must include the `logging=true` option within your connection string.

Installation

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

Configuration

If the application does not have an `app.config` application configuration file, then make one.

To enable Query Analyzer functionality, register the trace listeners in the `System.Diagnostics` section of the `app.config` file. The following example shows the format of a typical configuration file:

```xml
<system.diagnostics>
  <sources>
    <source name="mysql" switchName="SourceSwitch"
      switchType="System.Diagnostics.SourceSwitch">
      <listeners>
        <add name="EMTrace" type="MySql.ETrace.EMTraceListener, MySql.MonitorPlugin"
          initializeData="" 
          Host="SERVERHOST:SERVERPORT"
      </add>
    </source>
  </sources>
</system.diagnostics>
```
Within the configuration, set the values of the following parameters:

- **Host**
  
  The hostname and port number, separated by a colon, of the MySQL Enterprise Service Manager that receives the Query Analyzer data.

- **PostInterval**
  
  Query analyzer information is collected and then transmitted ("posted") in a batch from your application to the MySQL Enterprise Service Manager. This value specifies the number of seconds between each transmission. Choose this value carefully. Too long and it might take some time for queries to appear in the Query Analyzer.

- **UserId**
  
  The name of a user within MySQL Enterprise Service Manager that has rights to send agent information.

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

To get extended information on queries and have that information available through the MySQL Enterprise Monitor User Interface, enable the Connector/Net usage advisor. The extended information identifies potential issues such as a query not using an index, or not accessing all columns from a result set.

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

During execution of the application during development within Visual Studio, a significant amount of output is displayed in the Output window. To view this same trace output when running the application outside Visual Studio, configure an additional listener by adding the following within the `system.diagnostics` section of your `app.config` file:

```xml
<trace autoflush="false" indentsize="4">
  <listeners>
```

Usage

After you set up MySQL Enterprise Plugin for Connector/Net, you monitor the performance of your .NET applications through the **Query Analyzer** tab, as described in [Section 23.3, “Query Analyzer User Interface”](#).
Part III Using MySQL Enterprise Monitor
# Table of Contents

13 The User Interface ........................................................................................................... 129
  13.1 General ......................................................................................................................... 129
    13.1.1 Status Summary ................................................................................................. 129
    13.1.2 Asset Selector ...................................................................................................... 130
  13.2 The Query Analyzer ................................................................................................... 132
  13.3 What’s New .................................................................................................................. 132
14 The Dashboards ................................................................................................................. 135
  14.1 Introduction .................................................................................................................. 135
  14.2 Overview ..................................................................................................................... 135
  14.3 Replication ................................................................................................................... 138
  14.4 MySQL Instances Dashboard ..................................................................................... 140
    14.4.1 MySQL Instance Dashboard UI ........................................................................... 140
    14.4.2 MySQL Instance Details ..................................................................................... 142
    14.4.3 Adding Instances ............................................................................................... 144
    14.4.4 Filtering MySQL Instances ................................................................................. 146
15 Settings .............................................................................................................................. 149
  15.1 Global Settings ............................................................................................................ 149
  15.2 Manage Users ............................................................................................................. 155
  15.3 Logs ............................................................................................................................. 156
  15.4 User Preferences ........................................................................................................ 158
  15.5 Diagnostics Report .................................................................................................... 159
16 Reports and Graphs .......................................................................................................... 161
  16.1 All Timeseries Graphs .............................................................................................. 161
  16.2 InnoDB Buffer Pool Usage ....................................................................................... 162
17 Events ............................................................................................................................... 165
  17.1 Closing an Event ......................................................................................................... 167
  17.2 Notification of Events ............................................................................................... 168
18 Event Handling ................................................................................................................. 171
  18.1 Event Handling Page ................................................................................................. 171
    18.1.1 Event Handlers List ............................................................................................. 171
    18.1.2 Email Notification Group Controls .................................................................... 172
    18.1.3 Email Settings ................................................................................................... 174
    18.1.4 Email Notification Status .................................................................................. 174
    18.1.5 SNMP Settings .................................................................................................. 174
    18.1.6 SNMP Notification Status ................................................................................. 175
  18.2 Creating Event Handlers ............................................................................................. 175
    18.2.1 Event Action Log ............................................................................................... 177
    18.2.2 Suspending an Event Handler ............................................................................ 178
19 Advisors .......................................................................................................................... 179
  19.1 Advisors Page ............................................................................................................. 179
  19.2 Advisor Types ............................................................................................................ 183
  19.3 Advisor Thresholds ................................................................................................... 184
  19.4 Advisor Schedules ................................................................................................. 185
20 Expression-Based Advisor Reference ............................................................................. 187
  20.1 Administration Advisors ......................................................................................... 187
  20.2 Agent Advisors .......................................................................................................... 193
  20.3 Availability Advisors ............................................................................................... 193
  20.4 Cluster Advisors ...................................................................................................... 195
  20.5 Memory Usage Advisors ......................................................................................... 196
  20.6 Monitoring and Support Services Advisors ............................................................ 198
  20.7 Operating System Advisors ..................................................................................... 199
  20.8 Performance Advisors ............................................................................................. 199
  20.9 Replication Advisors ............................................................................................... 204
  20.10 Schema Advisors ................................................................................................. 209
  20.11 Security Advisors .................................................................................................. 213
Chapter 13 The User Interface

Table of Contents

13.1 General .......................................................................................................................... 129
  13.1.1 Status Summary ....................................................................................................... 129
  13.1.2 Asset Selector .......................................................................................................... 130
13.2 The Query Analyzer ....................................................................................................... 132
13.3 What’s New .................................................................................................................... 132

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

This chapter describes the user interface.

13.1 General

This documents global features of MySQL Enterprise Monitor User Interface, such as the Status Summary widget and the Asset Selector.

Note
Pressing Tab (forward) and Shift + Tab (back) will move the cursor's focus in the UI. Pressing Enter will make the selection.

13.1.1 Status Summary

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

The Status Summary:

Figure 13.1 A Status Summary Example

The Status Summary is displayed on every page, updates dynamically, and shows current status counters for:

- **Hosts monitored**: Local Agents are installed and collecting operating system related data.
- **MySQL Instances monitored**: Agent connections are successfully configured, and Instances are actively monitored.
- **MySQL Instances with invalid connection configurations**: Agent configurations exist but are not yet valid; correct or delete these connections on the MySQL Instances Dashboard.
- **Unmonitored MySQL Instances**: These running mysqld processes were detected by one or more Agents; configure connections to begin monitoring or ignore them on the MySQL Instances Dashboard.
- **Emergency Events**: These high priority problems require your immediate attention.
The Status Summary is global, so it does not take into account current filters such as Groups or selections in the Asset Selector. Therefore, a particular page may display results that differ from the global Status Summary overview.

The Status Summary counters are live links that link each value to its appropriate MySQL Enterprise Monitor User Interface page. Click the links to reveal additional details and tools to help resolve reported issues.

13.1.2 Asset Selector

MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

MySQL Enterprise Monitor 3.0 features a new Asset Selector on the left side of many UI pages that controls content in the main-body of the page. On the Events page, for example, selecting a Host, Group, or MySQL Instance displays only those Events relevant for the selected Asset. Reports & Graphs pages also feature a full-blown Asset Selector, while other UI pages show or hide various Asset types depending on the content of the page.

The Asset Selector content is automatically populated as monitoring connections are made and maintained and state information is gathered. The Groups shown in the Asset Selector are the ones you create and manage (on the MySQL Instances dashboard or during Agent installation), and those that are discovered automatically from monitoring replication topologies.

Controls

Use the filter icon at the top of the Asset Selector panel to open a control panel:

The Show All Assets checkbox. Toggle showing only Groups and MySQL Instances ("streamlined" mode) or showing all known Assets. These include:

- Hosts
  
  This include hosts with (and without) MySQL instances.

- Host-related Assets such as File Systems and Network Interfaces
- Monitoring-related Assets such as:
  
  MEM Agents
  
  The Service Manager itself, and its related Assets
  
  The Tomcat web server, used by the Service Manager
- And your monitored MySQL Instances and Groups

In "stream-lined mode", MySQL Instances are directly contained in Groups. In "All" mode, Groups contain Hosts, which then contain the MySQL Instances running on those Hosts; the same Group membership is represented but with a bit more detail about the operational relationship. Agents are also shown and listed as subordinate to the Hosts that they are running on.

Note

The Query Analyzer page does not allow the selection of the "All" mode, as Query Analyzer only applies to Groups of MySQL Instances.
• A Search box. Search all Assets for the specified Display Name; this provides a quick way to navigate to a specific MySQL Instance or Host when you know the name and is especially useful if you are monitoring a large number of Assets.

The search is case-insensitive, and requires the search string to be 3+ characters in length. Searching for nothing (blank) resets the search filter.

• Expand All / Collapse All tree controls.

Note

Expanding this can be expensive when you are monitoring a large number of Assets.

There is also a control to horizontally collapse the entire Asset Selector against the left-hand side of the page and allow the main-body content to take more screen width.

Asset Naming

Most Assets names are automatically derived from the conventional name and other identifying details of the Asset. For example, MySQL Instances are displayed as the "[hostname]:[port]" of the instance unless you have manually edited the Display Name on the MySQL Instances page.

Network interfaces display as: "[network-interface name] ([network address])", while File Systems are shown as: "[mount-point name] ([filesystem type])".

Agents are listed as "Agent ([UUID-fragment])", where "UUID-fragment" is the first and last 3 characters of the Agent's internal UUID (the remaining 26 characters are omitted); these 6 characters should be sufficiently unique to help you identify the right Agent when the UI presents you with a selection list.

Finally, the Service Manager has a built-in Agent that is displayed as "MEM Built-in Agent".

Asset Styling and Annotations

Asset Status Styling

• Agent Unavailable: The Asset Selector displays Assets for which it has no recent status information in red italic text. This usually means the Agent responsible for monitoring the Asset is unexpectedly not communicating to the Service Manager.

• Asset Unavailable: An Asset which is actively monitored but determined by its Agent to be inaccessible ("down") is displayed in red text.

• Planned Notification Blackout: An Asset which is in a planned "Blackout" state with respect to Event Notifications is displayed in gray italic text.

• Logical Assets: An Asset which is inferred to be present and needed as a "container" but not actually being monitored is displayed in gray italic text. If you are remotely monitoring an MySQL Instance, for example, its Host would be displayed in gray italic text.

Emergency Event Badges

The Asset Selector displays a numeric badge next to any Asset that has open Emergency Events, and the text of the asset changes colour to red. Click on the badge to view the Emergency Events. These Emergency Events apply to assets which are monitored by Advisors checking for those specific conditions, only. For example, if one of the network interfaces becomes unavailable, it is not marked with a badge, it is removed from the Asset Selector because the network-specific Advisors do not check for the availability of network interfaces, just the data transmitted on them. For assets such as agents, filesystems and MySQL instances, Advisors monitor their availability in addition to their activities. If an agent or instance becomes unavailable, or a filesystem approaches its free-space limit, a numeric badge is displayed on the asset's icon and the name of the asset is displayed in red.
If problems exist for multiple assets on a host, the number of problems on that host are displayed in a numeric badge at the host level.

MySQL Instance Notes

If you annotate a MySQL Instance with a Note on the MySQL Instances page, the Asset Selector displays a Note Icon to the right when you hover over the Instance. View the Note in a tooltip by then hovering over the Note Icon.

13.2 The Query Analyzer

The MySQL 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, removing unique data such as literal values, so that the generic queries can be counted and identified more easily.

Query Analyzer works by collecting information about SQL statements that your MySQL client application sends to the MySQL server. The built-in PS feature with MySQL Server 5.6 and greater is preferred and simplest to set up, but other options include the MySQL Enterprise Monitor Agent proxy service, or by using the MySQL Enterprise Monitor Plugin for a given Connector that sends the information directly to MySQL Enterprise Service Manager.

For more information about using the Query Analyzer, including how to provide queries to MySQL Enterprise Service Manager, how to use the user interface, and how to drill down to find the queries and information that you need, see Chapter 23, Using the Query Analyzer. For the meaning of each field on the Query Analyzer tab, see Section 23.3, "Query Analyzer User Interface".

13.3 What's New

The What's New page provides a simplified interface for providing updates, information about what to do next, and news related to your My Oracle Support account. By default, the information is automatically updated every hour.

Access the What's New from the help menu in the top navigation bar, and select What's New?.

Figure 13.2 The What’s New Page

The What's New page is divided into three main sections:
Welcome to MEM

• Welcome to MEM, including sections for New users and Existing users
• My Oracle Support
• Important Product-Related Announcements

Welcome to MEM

On the left-hand side of the page, this section provides information about your setup, what you may do next to complete the set up, and related information. Sections exist for new and upgraded installations.

For related information, see Chapter 6, Post-installation Considerations.

My Oracle Support

My Oracle Support is the channel for interacting with Oracle Support across all products. MySQL Enterprise Monitor (MEM) provides a basic integration with My Oracle Support (MOS), allowing you to see an up-to-date list of Open and waiting-on-customer support issues in the product.

Setup

In Global Settings, provide your own My Oracle Support / Oracle SSO credentials. These are stored encrypted in the MEM repository, and are used to sign in to the MOS services. For more information about these settings, see Section 15.1, “Global Settings”.

Operation

MySQL Enterprise Monitor will, in the background, periodically check the status of MOS Service Requests (SRs) associated with the account with the provided credentials.

Any currently "Open" SRs will be listed on the What’s New page.

Any Open SRs that are in a "Waiting on Customer" state - those that need your attention - will also be present on the Overview Dashboard.

In both locations, the SRs will be linked to the My Oracle Support system for your convenience.

Configuration

Use the Configure What’s New button on the What’s New page to control if the MOS check should be performed, and also to force an immediate update of the status of MOS Support Requests.

Note

To poll the My Oracle Support site directly, your MySQL Enterprise Service Manager must have access to the My Oracle Support website. If your MySQL Enterprise Service Manager is not directly connected to the Internet, you can set an HTTP proxy to help access the information. For more information on configuring an HTTP proxy, see Section 15.1, “Global Settings”.

Important Product-Related Announcements

This section provides a list of news items from a Web-based news feed. The news feed contains important information, including, but not limited to:

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

To configure the operation and forcibly update the information on the What's New page, click the Configure What's New button at the top-left of the What's New page.

Figure 13.3 What's New Configuration

The configurable options are:

- **Automatically check for Product-Related Announcements?**

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

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

  You can force an update of the New Announcements information by selecting the yes radio button next to Force check now?. A check will be executed when you click the save button to close the configuration dialog.

- **Automatically check for updates to open My Oracle Support Issues?**

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

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

  You can force an update of the Support Issues information by selecting the yes radio button next to Force check now?. A check will be executed when you click the save button to close the configuration dialog.

Click save to save your settings. If you selected any system to force an update, the update will take place in the background. To cancel your changes, click cancel.
Chapter 14 The Dashboards

Table of Contents

14.1 Introduction ......................................................... 135
14.2 Overview ............................................................ 135
14.3 Replication ............................................................ 138
14.4 MySQL Instances Dashboard ...................................... 140
   14.4.1 MySQL Instance Dashboard UI .............................. 140
   14.4.2 MySQL Instance Details ................................. 142
   14.4.3 Adding Instances ........................................... 144
   14.4.4 Filtering MySQL Instances .............................. 146

This chapter describes the MySQL Enterprise Monitor User Interface dashboards.

14.1 Introduction

The following dashboards are provided:

- Section 14.2, “Overview”: displays the high-level status of your MySQL databases and hosts.
- Section 14.3, “Replication”: displays the replication topology and current state of your replication servers.
- Section 14.4, “MySQL Instances Dashboard”: displays the MySQL instances currently monitored and enables you to add, remove, and group instances.

14.2 Overview

The Overview Dashboard shows a high level summary of the MySQL instances and hosts that are monitored by MySQL Enterprise Monitor.

The Overview summarizes various key statistics related to the group that is selected, such as Database Availability, Connections, Database Activity, Query Response Times, and any current MySQL Instances or Hosts that have active Critical or Emergency level events against them. It is designed in this way to give you a quick high level picture of assets that require immediate attention, as well as give you an up to date profile of how MySQL Instances within environment are behaving.

Group Overview Configuration

The Group Overview configuration bar enables you to define the default group to view in the Overview dashboard. It also enables you to define the period for which the data is displayed.

By default, the Group Overview is set to All. To edit the overview, click Edit. The Group Overview edit bar is displayed.

Figure 14.1 Group Overview Configuration

The controls in the Group Overview configuration bar are described in the following table:

Table 14.1 Group Overview Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Overview</td>
<td>Drop-down list containing all groups defined in MySQL Enterprise Service Manager.</td>
</tr>
</tbody>
</table>
Database Availability

### Graph Time Range

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graph Time Range</strong></td>
<td>Drop-down lists containing the time periods to apply to the graphs. The possible values are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Interval</strong>: select the duration for the overview data. If you select 1 hour, the data collected in the last hour is displayed.</td>
</tr>
<tr>
<td></td>
<td>• <strong>From/To</strong>: select a date and time range for the overview data.</td>
</tr>
<tr>
<td>Go</td>
<td>Applies the new configuration to the Overview dashboard.</td>
</tr>
<tr>
<td>Save as Default</td>
<td>Sets the selected group and time range as the default.</td>
</tr>
<tr>
<td></td>
<td>It is not possible to save a date range, using From/To, as the default for a group.</td>
</tr>
<tr>
<td>Reset</td>
<td>Resets the group overview configuration to the previously saved values.</td>
</tr>
</tbody>
</table>

**Important**

If the default group is deleted, an error is displayed on the Overview dashboard.

### Database Availability

The Database Availability statistics show an aggregate of availability statistics, generated by the MySQL Availability Advisor, for all MySQL Instances within the selected Group, and enables you to monitor your Service Level Agreements for availability.

**Note**

The MySQL Availability Advisor must be enabled for this functionality to work. It is enabled by default.

Database Availability is computed by each individual MySQL Enterprise Monitor Agent attempting to make a connection to each monitored MySQL Instance on a frequent basis (by default every 1 second, but this is configurable within the Advisor), to check whether the MySQL Instance is actively accepting new connections.

Depending on the response from the MySQL Instance, the availability is broken down in to four categories:

- **Available**: The instance was actively monitored, and we could make a new connection.
- **Up**: The instance Uptime status variable reported it was up when we last logged in, but we were not actively monitoring at this time. This can happen when you start monitoring for the first time from an Agent, or if the Agent was down for some period of time.
- **Unreachable**: The Agent could not get a response from the MySQL Instance, however when we logged back in we detected that the instance was up for that period of time from the Uptime status variable. This can happen for various reasons, such as a max_connections limit being reached, or some networking problem between the MySQL Enterprise Monitor Agent and the MySQL Instance that is being monitored.
- **Down**: The Agent could not get a response from the MySQL Instance at all.

For each time range, the bar chart is split up in to slices of time (1 or 2 pixels each, depending on the width of the display), which represent a period of time; Day = 4 minutes, Week = 20 minutes and Month
Overview Graphs

= 2 hours. Within those slices, the time within each availability state is aggregated across all MySQL Instances within the selected Group, and shown as the percentage of the total time in the slice.

For example, if you have four MySQL Instances within the selected group, with three of them being up and one of them being down for a 4 minute period within the day, the slice representing those 4 minutes would be 75% dark green, and 25% red.

Note
MySQL Availability reporting is only available when using a MySQL Enterprise Monitor Agent of version 3.0.0 and above.

Overview Graphs

The following graphs are displayed by default:

- Connections - All MySQL Instances
- Database Activity - All MySQL Instances (Always displayed.)
- Query Response Time Index

The graphs are customizable. To change a graph, do the following:

1. Select the title of the graph. The graph selection drop-down list is displayed.
2. Select the required graph from the drop-down list. The graph updates to your selection and is saved for future sessions.

To move the graphs, use the move icon in the top right corner of the graph.

It is also possible to add graphs. To add a graph, do the following:

1. Click the Add a new graph button. The graph selection drop-down list is displayed.
2. Select a graph. The page reloads and the selected graph is displayed.

To remove a graph, click the red X in the top-right corner of the graph. The page reloads and the selected graph is removed.

Current Problem MySQL Instances and Hosts Panels

The Current Problem MySQL Instances and Current Problem MySQL Hosts list the top 5 MySQL Instances and Hosts respectively, based on whether they have open events with a current status of either Emergency or Critical.

The results are sorted by the total time that each event has been within these statuses, by Emergency descending, and then Critical descending. This means that the MySQL Instances, or Hosts, that have had Emergency, then Critical events open for the longest will float to the top of the list.

This is extremely useful in environments that are monitoring many MySQL Instances, as it enables you to focus in to which MySQL Instances or Hosts to tackle next, by telling you those that have the most problems against them.

When a MySQL Instance or Host has any open Emergency events against them, the row for that asset is highlighted in red.

Current Emergency & Critical Events

The Current Emergency & Critical Events panel lists a stream of the currently open events with an Emergency or Critical status. These are listed separately to the Current Problem MySQL Instances or Current Problem Hosts panels, as these might not show all assets within an environment that is monitoring five or more of either asset type.
This enables you to see all current high priority events within a single panel for all monitored assets.

14.3 Replication

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

Navigate to the Replication page by choosing Replication under Dashboards. This page summarizes the state of your replication servers; you can drill down to see details about any master or slave. Using this page helps you avoid running the SHOW SLAVE STATUS command over and over on multiple servers; for consistency, the Replication page uses some of the same keywords as the output from that command.

Note
Set up agents to monitor each master and slave server. Only servers that are monitored appear on this page.

The Replication page groups all master servers with their slaves. Masters and their slaves are autodiscovered and a grouping is created, based on the way that the servers are interconnected (known as the replication topology). 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.

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 for debugging the replication topology discovery process.

Warning
Auto-discovery with remote monitoring only functions with MySQL 5.6 and later. Earlier versions of MySQL server require the Agent to be installed on the same host as the monitored MySQL instance. This is because mysqld did not expose the master’s uuid value via SHOW SLAVE STATUS until version 5.6.

You can manage replication groups from the MySQL Instances page in the same way as other groups. However, any slaves removed from a server group are automatically restored to that group. You can also add non-slaves to a replication grouping.

Note
The replication group name can also be edited from the Replication dashboard by clicking the drop-down menu for the group, and choosing Rename Group.

Replication Page Details

Choose a value from the Refresh drop-down list box to set the rate at which information is updated. This refresh rate applies only to the information presented on this page: It is independent of the rate set for the Monitor tab.

The following columns describe replication master and slave servers:

- **Servers**: Displays the group name and any servers that are part of the group. Levels of indentation in this column show the relationship between master servers and their slaves. The icon next to each server indicates if the server is enabled for semi-synchronous replication or not. A gray “disabled”-style icon indicates that semi-synchronous replication is not available.

- **Type**: Indicates the topology of a server group or in the case of individual servers, whether a server is a master, a combined master/slave, or a slave.
• **Threads**: Displays information about the two dedicated replication threads that run on the slave server. Both threads must be running for the slave to work properly. **IO** reports the status of the slave I/O thread. **SQL** reports the status of the slave SQL thread.

• **Time Behind**: The interval that the slave is behind the master, in **hh:mm:ss** format. This column is blank if a server is a master. Typically, you try to keep this value low.

• **Binary Logs**: Displays information about the **binary log** file. **Current File** shows the binary log file name. **Position** shows the current position in the binary log file.

• **Master Position**: Displays information about the binary log position from the master server. **Binary Log** shows the master binary log file name. **Position** shows the current position in the master binary log file.

• **Log Space**: Displays the sizes of log files related to replication. **Binary Logs** shows size information for the binary log files. **Relay Logs** shows size information for the relay log files.

Most column headings are active links; click the header to change the display order. Sorting works differently for different column groupings. Click the **Time Behind** header to order servers by the number of seconds they are behind their master. The server topology is respected regardless of sort order. For example, in a **TREE** topology, ordering occurs within branches only.

If the **Agent** is down, instances 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, the last know status of the IO or SQL threads is shown in italics.

Clicking a master server opens a dialog box that displays information about the server. The information shown includes:

- The number of slave servers.
- The binary log file name.
- The binary log position.
- Which databases are replicated and which not.
- GTID values, for MySQL servers 5.6 and above.

The dialog box also includes a link to hide or show the slave servers.

Clicking a slave server opens a dialog window showing extensive information about the slave.

### Deleted Replication Groups

This section describes the how replication groups are treated if the replication topology changes.

- If all instances which make up a replication group are deleted, the replication group is deleted automatically.

  If a replication group is automatically deleted, an event is generated and a message is displayed in the MySQL Enterprise Monitor User Interface.

- If the deleted replication group was associated with an event handler, the replication group is automatically removed from the event handler's definition.

  An event is generated and a message is displayed in the MySQL Enterprise Monitor User Interface stating that the group was removed.

- If the deleted replication group was the only filter defined on the event handler, the event handler is suspended.

  An event is generated and a message is displayed in the MySQL Enterprise Monitor User Interface stating that the group was removed and the event handler suspended.
Usage Notes

For information on the kinds of problems you might find while monitoring replication, and how to solve them, see Troubleshooting Replication and Improving Replication Performance.

14.4 MySQL Instances Dashboard

The MySQL Instances dashboard presents information on the current connection status of all monitored instances. It also presents information on unsuccessful connections, uncontactable agents, and MySQL instances which are not yet monitored.

The MySQL Instances Dashboard contains the following:

- **MySQL Instance Details**: this list is displayed by default. It lists all the currently monitored instances. If there is a problem with a connection to one of those instances, it is highlighted in red.

  For more information, see Section 14.4.2, “MySQL Instance Details”.

- **Unreachable Agents**: displayed only if a previously contactable agent is no longer contactable.

  For more information, see Unreachable Agents.

- **Bad MySQL Connections**: displayed if misconfigured connections exist.

  For more information, see Bad Connection Configurations.

- **Unmonitored MySQL Instances**: lists the number of MySQL instances which are available, but currently unmonitored by MySQL Enterprise Monitor.

  For more information, see Unmonitored MySQL Instances.

14.4.1 MySQL Instance Dashboard UI

This section describes the MySQL Instances Dashboard.

At the top of the dashboard are the following buttons:

- **Create Group**: opens the Create Group dialog. This enables you to create an empty MySQL instance group that will immediately be displayed on the MySQL Instance Details listing. Groups are used to organise your monitored instances.

- **Add MySQL Instance**: enables you to add a MySQL instance for MySQL Enterprise Monitor to monitor.

- **Add Bulk MySQL Instances**: enables you to add multiple MySQL instances for MySQL Enterprise Monitor to monitor, all of which use the same configuration details.

**Alert Buttons**

The alert buttons list the number of problematic instances, connections, and agents in your implementation. If a problem exists, they are displayed on the top-right side of the dashboard.

- **Unmonitored MySQL Instances**: lists the number of MySQL instances which are available, but currently unmonitored by MySQL Enterprise Monitor. Click to open the Unmonitored MySQL Instances list. See Unmonitored MySQL Instances for more information.

- **Bad MySQL Connections**: displays the number of misconfigured connections to MySQL instances. Click to open the Bad Connection Configurations list. See Bad Connection Configurations for more information.

- **Unreachable Agents**: lists the number of agents which are currently uncontactable. Click to open the Unreachable Agents list. See Unreachable Agents for more information.
Bad Connection Configurations

This section lists the connection configurations which are unable to establish a connection with the MySQL instance.

Table 14.2 Bad Connection List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Host</td>
<td>Lists the hostname of the monitoring agent.</td>
</tr>
<tr>
<td>Connection Details</td>
<td>Lists the IP address defined in the connection string.</td>
</tr>
<tr>
<td>Last Error Date</td>
<td>Date and time of the last occurrence of this error.</td>
</tr>
<tr>
<td>Error Details</td>
<td>Cause of the error.</td>
</tr>
</tbody>
</table>

If the bad connection results from a misconfiguration, select **Edit Connection** from the drop-down menu. The connection configuration window is displayed, enabling you to review and edit the connection.

To delete the bad configuration, select **Delete Connection** from the drop-down menu.

Unreachable Agents

This section lists the agents which are configured, and were communicating with the Service Manager, but cannot be contacted.

Table 14.3 Unreachable Agents List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Hostname of the server on which the agent is installed.</td>
</tr>
<tr>
<td>State</td>
<td>State of the agent. For example, if the agent is shut down properly, it signals the Service Manager that it is shutting down, and the state is displayed as <strong>SHUTDOWN</strong>. If the agent did not shutdown properly, if its host shutdown unexpectedly, or due to a network fault, the state displayed is <strong>TIMEDOUT</strong>.</td>
</tr>
<tr>
<td>Last Seen</td>
<td>Time and date at which the agent last contacted the Service Manager.</td>
</tr>
<tr>
<td>Version</td>
<td>Agent version.</td>
</tr>
<tr>
<td>UUID</td>
<td>The unique identifier of the agent.</td>
</tr>
<tr>
<td>Agent Directory</td>
<td>Agent installation directory.</td>
</tr>
</tbody>
</table>

Unmonitored MySQL Instances

This section lists the running MySQL instances which have been detected but not added to the system. You can monitor, ignore, or cancel these connections if pending.

To begin monitoring one, or more, of the unmonitored instances, select them using the checkboxes and click **Monitor Instances**. For more information on adding connections, see Section 14.4.3.1, “Adding a MySQL Instance”.

To ignore instances, make your selection and click Ignore Instances. A checkbox, **Display n ignored instances** is displayed, where n is the number of instances ignored. To undo the ignore, and display the instance, check the **Display n ignored instances** checkbox, select the instance and click **Show Instance**.

If no unmonitored instances are present, the ignored instances are listed instead.

---

Important

If an ignored instance is uninstalled, the ignored instance is removed from the list of unmonitored instances.
To cancel a pending connection, select the pending connection and click **Cancel Pending Connections**.

Table 14.4 Unmonitored MySQL Instances List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>The server on which the running MySQL instance was discovered.</td>
</tr>
<tr>
<td>Connecting</td>
<td>Whether a connection is being attempted with the instance.</td>
</tr>
<tr>
<td>Port/Socket</td>
<td>Port or socket on which the MySQL instance is listening.</td>
</tr>
<tr>
<td>Process ID</td>
<td>The process ID of the running instance.</td>
</tr>
<tr>
<td>Process User: Group</td>
<td>ID of the user and group.</td>
</tr>
<tr>
<td>Process Arguments</td>
<td>The arguments with which the instance was started.</td>
</tr>
</tbody>
</table>

### 14.4.2 MySQL Instance Details

The **MySQL Instance Details** section lists all the instances currently monitored by this installation and enables you to delete and edit instance configuration.

**Editing Instances**

To edit an instance, do the following:

- Select an instance by selecting the checkbox on the left of the instance name.
- Click Edit Instances. The Edit Instances dialog is displayed.

The Edit Instances window is identical to the Add Instance Connection window described in Section 14.4.3.1, “Adding a MySQL Instance”, with the exception of the first tab, Instance Details.

When editing an individual instance, the **Instance Details** tab enables you to edit the instance Display Name and add notes on the instance. For example, if the instance name is MySQLServer001, and ThisIsMyServer is added in the Display Name field, ThisIsMyServer is displayed in the MySQL Instance Details list, and everywhere else the instance name is used.

If you add a note, a note icon is displayed in the Notes column for that instance.

**Deleting Instances**

To delete an instance, or multiple instances, select the instance(s) and click **Delete Instances**, or select **Delete Instance** from the instance-specific drop-down menu.

**Columns**

The following columns are available:

Table 14.5 MySQL Instance Details Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>The instance names, in their assigned Groups. If no groups are defined, all MySQL instances are contained by the All group. The checkbox enables you to select all instances.</td>
</tr>
<tr>
<td>Notes</td>
<td>Displays a note icon, if a note was defined on the Instance Details tab. If a note was defined, hover the cursor over the note icon. The note is displayed as a tooltip.</td>
</tr>
</tbody>
</table>
MySQL Instance Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Versions: MySQL</td>
<td>Displays the version of the monitored MySQL instance.</td>
</tr>
<tr>
<td>Versions: Agent</td>
<td>Displays the version of the monitoring agent.</td>
</tr>
<tr>
<td>Versions: Operating System</td>
<td>Displays the type and version of operating system on which the MySQL instance is installed.</td>
</tr>
<tr>
<td>Port</td>
<td>Displays the configured MySQL port.</td>
</tr>
<tr>
<td>Data Dir</td>
<td>Displays the configured data directory of the MySQL installation.</td>
</tr>
</tbody>
</table>

Group and Instance Context Menu

The group-level context menu contains the following:

- **Add to Group**: Adds a MySQL instance to the group.
- **Delete Group**: Removes the group (the instances within the group will not be removed, but will be dissociated from the removed group).
- **Rename Group**: Renames the group.

**Note**
The text is captured as is in the text field. For example, HTML entities are not converted.

- **Support Diagnostics**: Opens the Support Diagnostics page. This enables you to generate a set of reports which you can send to MySQL Support as an attachment to a reported issue. This report can take several minutes to generate. The reports archive also includes a SQL dump of the Advisor Schedules, Inventory and Configuration schemas.

**Important**
The Configuration schema may contain login credentials. However, these credentials are encrypted using keys which are not stored in the repository and are not included in the Support Diagnostics report.

**Important**
Because generating a the Support Diagnostics report is an expensive operation, the report is cached for six hours. All requests within this six hour time period will pull (download) this cached report. A request after this period will trigger a new report to be generated.

Figure 14.2 MySQL Instances: Group Context Menu

**Note**
The context menu for the default group, **All**, only contains the **Support Diagnostics** menu item. It is not possible to edit the **All** group.

The instance-level menu contains the following:

- **Remove from Group**: removes the selected instance from the group. This menu item is present only if the instance is a member of a group.
• **Edit Instance**: opens the Edit Instance dialog.

• **Delete Instance**: deletes the instance from the MySQL Enterprise Service Manager.

<table>
<thead>
<tr>
<th>Important</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is not possible to delete the MySQL Enterprise Monitor repository from the list. If you delete it, it is automatically restored to the list.</td>
</tr>
</tbody>
</table>

• **Refresh Inventory**: forces an inventory of the selected instance.

• **Support Diagnostics**: opens the Support Diagnostics dialog.

• **Enable Event Handler Blackout**: stops all Event Handlers associated with the selected instance. Events continue to be generated and advisors continue evaluating the data collected by the agent monitoring the selected host, but all event handlers are suspended for the selected instance.

### 14.4.3 Adding Instances

This section describes how to add MySQL Instances to MySQL Enterprise Monitor. The following topics are described:

- **Section 14.4.3.1, “Adding a MySQL Instance”**
- **Section 14.4.3.2, “Adding Multiple MySQL Instances”**

#### 14.4.3.1 Adding a MySQL Instance

This section describes how to monitor a MySQL instance.

**Connection Settings**

The Connection Settings tab

**Figure 14.3 Add Instance Connection Settings**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor From</td>
<td>Select an Agent from the list of Agents to monitor this MySQL database Instance.</td>
</tr>
</tbody>
</table>
Adding Instances

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>It is recommend installing one Agent per Host and using that Agent to monitor all its MySQL Instances.</td>
</tr>
<tr>
<td></td>
<td>Set up the Agent as a local connection by specifying TCP/IP and an <strong>Instance Address</strong> of 127.0.0.1, or use a socket file. If there is no local Agent on the Host and you are unable to install one, use the built-in or another Agent to monitor remotely.</td>
</tr>
</tbody>
</table>

**Note**
If the instance is monitored remotely, it is not possible to retrieve any information on the host. To monitor a host, an agent must be installed on that host.

**Connect Using**
Select TCP/IP or a socket to connect to the instance. Socket connections can only be used for an Agent that is installed on the same machine as the target instance, and do not work with instances running on Windows.

**Instance Address and Port**
The IP address, or valid hostname, and port number the instance is listening on. If the host/agent chosen is local to this instance, you should use 127.0.0.1 here.

**Admin User and Password**
The root user, or user with the privileges defined in Creating the **Admin User** and the password.

**Auto-Create Less Privileged Users**
Choose Yes to create the General and Limited users on the MySQL instance. You must add a user name and password for both. For more information on these users, see Section 5.2, “Creating MySQL User Accounts for the Monitor Agent”.

Choose No if you intend to use the Admin user for all data collection.

**Note**
It is strongly recommend to use the General and Limited user.

**General User and Password**
Add a user name and password for the General User.

**Limited User and Password**
Add a user name and password for the Limited User.

### Advanced Settings

**Figure 14.4 Add Instance Advanced Settings**

![Figure 14.4 Add Instance Advanced Settings](image)
### Table 14.7 Advanced Settings

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover Replication Topologies</td>
<td>Specifies whether the agent attempts to discover if the instance is part of a replication group and discover the other members of that replication group. When performing replication topology discovery, the agent attempts to read the slave's master.info, and use the stored credentials to log in to the master to read its inventory table and retrieve the master's UUID. If set to <strong>No</strong>, no replication discovery is attempted.</td>
</tr>
<tr>
<td>MySQL Identity Source</td>
<td>Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong>: uses either the <code>server_uuid</code> variable if present, or generates a random new UUID</td>
</tr>
<tr>
<td></td>
<td>• <strong>Host Plus Datadir</strong>: generates a hash of the host identity and the path to the MySQL instances data directory to create a unique identity.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td><strong>Host Plus Datadir</strong> can be used only if the agent is running on the same host as the MySQL instance for this connection.</td>
</tr>
<tr>
<td>Inventory Table Schema</td>
<td>When the Agent connects to the MySQL Instance, it creates an inventory table, if one does not already exist, and stores two rows within it: a generated Instance UUID, and the host ID. By default this is created within the <code>mysql</code> database. On shared hosts or cloud environments this may not be accessible to the Agent user; provide a database name to override where the inventory table is created.</td>
</tr>
<tr>
<td>Connection Timeout</td>
<td>Connection timeout, in milliseconds, used by the JDBC driver.</td>
</tr>
<tr>
<td>Socket Timeout</td>
<td>Socket timeout, in milliseconds, used by the JDBC driver.</td>
</tr>
</tbody>
</table>

### Group Settings

Enter the groups to which you want to add the instance. It is also possible to define new groups in this field.

### 14.4.3.2 Adding Multiple MySQL Instances

The **Add Bulk MySQL Instances** tabs are identical to those used to add a single instance, with the exception of the **Instance Address** field, which is replaced by the **Connection Endpoints** field in the bulk version. To add multiple MySQL instances, add the comma-separated list of MySQL addresses to the **Connection Endpoints** field in the format of `Hostname:PortNumber`.

To add the instances successfully, you must ensure the user credentials, encryption settings, and so on, are identical across all instances added.

### 14.4.4 Filtering MySQL Instances

To search for specific instances, click the filter icon. The MySQL Instance filter is displayed.
Filtering MySQL Instances

Figure 14.5 MySQL Instance Filter

Table 14.8 MySQL Instance Filter

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Name</td>
<td>Search on full or partial name of the server.</td>
</tr>
<tr>
<td>Server UUID</td>
<td>Search on the UUID of the server.</td>
</tr>
<tr>
<td>Server ID</td>
<td>Search on the server_id.</td>
</tr>
<tr>
<td>Query Analyzer</td>
<td>Search for servers on which the Query Analyzer is enabled, or not.</td>
</tr>
<tr>
<td>MySQL Version</td>
<td>Search for specific MySQL version numbers.</td>
</tr>
<tr>
<td>Agent Version</td>
<td>Search for specific MySQL Enterprise Monitor Agent version numbers.</td>
</tr>
<tr>
<td>Operating System</td>
<td>Search on the Operating Systems on which the server is installed.</td>
</tr>
</tbody>
</table>

The Agent Version and MySQL Version fields support the use of range operators (>, < =), enabling you to define ranges of versions to filter on. For example, setting MySQL Version to <=5.1 returns all MySQL instances older than MySQL 5.1.

Note

Filtering on MySQL or Agent version uses a regular expression which does not support the use of partial version numbers, such as "5.". 5 or 5.6 return a result, if such versions are in use, but a partial version returns an error.
Chapter 15 Settings

Table of Contents

15.1 Global Settings .................................................................................................................................................. 149
15.2 Manage Users ..................................................................................................................................................... 155
15.3 Logs ................................................................................................................................................................. 156
15.4 User Preferences ............................................................................................................................................. 158
15.5 Diagnostics Report ........................................................................................................................................ 159

This chapter describes the various configuration options available in the MySQL Enterprise Monitor User Interface.

15.1 Global Settings

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

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

The Global Settings page is divided into these sections:

• Server Locale

Figure 15.1 MySQL Enterprise Monitor User Interface Settings: 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 MySQL Enterprise Monitor User Interface is running.

• Server Hostname

You can alter the Hostname, Port Number, and Login Display Name used to identify the MySQL Enterprise Service Manager when reporting notifications.
Global Settings

Figure 15.2 MySQL Enterprise Monitor User Interface Settings: Server Hostname

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

• Data Purge Behavior

  The Data Purge Behavior section of the Global Settings page lets you remove old log files and also old data from the repository. The default purge interval is 4 weeks. To purge data, change this setting by choosing from the drop-down list. Choosing 12 months, for example, removes all data that is older than a year.

Figure 15.3 MySQL Enterprise Monitor User Interface Settings: Data Purge

Warning
Purging data permanently removes information from the repository. Since events are derived from data contained in the repository, they are purged along with the data.

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

Note
The purge process is started approximately once every day, or when the MySQL Enterprise Monitor User Interface is restarted. If you change the purge duration from a large timespan to a smaller one, the data will be purged
in increments of one hour, from oldest to newest, until the new data retention policy is met. This is done to reduce the load on the repository.

Note
The system assumes that events will be auto-closed, or closed manually. The purge functionality only purges closed events and related data.

You can configure the data purge behavior in the following ways:

- **Remove Historical Data Collection Older Than** configures the duration that the main data about your servers is retained. This includes all data collections, including CPU, memory and connections and activity statistics.

- **Remove Query Analyzer Data Older Than** configures the duration that the query analyzer statistics and information about individual queries is retained.

Notes for setting purge behavior:

- Purging can be carried out manually by enabling the `innodb_file_per_table` for the repository database and then using an `OPTIMIZE TABLE` operation to reclaim space from deleted rows in the table.

- If you change the purge value from a high value to a very low value, the space used by the purged data is not reclaimed from the InnoDB tablespaces. Do this by running `OPTIMIZE TABLE` on the MySQL tables for MySQL Enterprise Service Manager to reclaim the space from the purged rows.

- **My Oracle Support Credentials**

  You can specify the credentials for logging into the My Oracle Support site. These should match the user name and password that you have registered with Oracle for access to the support site.

  ![Figure 15.4 MySQL Enterprise Monitor User Interface Settings: My Oracle Support Credentials](image)

  **Note**
  Only administrators can change the **My Oracle Support Credentials** section; for other users, this section does not show up in the interface. For more information about different users and their rights see Section 15.2, “Manage Users”. Specifying incorrect credentials results in the error message, “Your credentials do not appear to be valid.”

- **HTTP Proxy Settings**

  You might want to update your HTTP Proxy Settings if your MySQL Enterprise Service Manager is not directly connected to the internet. The proxy settings are used when updating the information within the **What's New** tab. For more information, see Section 13.3, “What's New”.
Figure 15.5 MySQL Enterprise Monitor User Interface Settings: HTTP Proxy Settings

- **LDAP Authentication**

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

The configurable elements for LDAP authentication are:

- **Use LDAP for Authentication**

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

- **LDAP is Authoritative**

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

- **Primary Server Hostname**

  Hostname or IP address of the primary LDAP directory server.

- **Port Number**

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

- **Secondary Server Hostname (optional)**

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

- **Connect timeout:**

  Time elapsed without establishing a connection to the LDAP server. If a connection is not established within the defined number of seconds, an error is returned.

  **Note**

  If the value is set to 0, the **Connect timeout** value defaults to the operating system's default value.

- **Read timeout:**

  Time elapsed without a response to a request for data from the LDAP server. If no response is received within the defined number of seconds, an error is returned.
Global Settings

Note

If the value is set to 0, the Read timeout value defaults to the operating system's default value.

- Port Number

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

- Encryption

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

- Referrals

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

- LDAP Server Allows Anonymous Binds

Optionally allow Anonymous binds.

When unchecked (disabled), MySQL Enterprise Monitor provides for a pre-auth bind user to lookup account records. On "Active Directory," the most common user account attribute is "sAMAccountName", whereas it is common for Unix based LDAP to use CN. If the Active Directory server is not configured to honor CN binds, then it will fail to fetch credentials.

- Authentication Mode

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

- Update Password On Save box and password fields

To enter the LDAP server password, check the Update Password On Save box first. When you return to this dialog to update settings other than the LDAP password, leave this box unchecked to avoid blanking out the saved password. If the LDAP password does change later, check the box again and enter the new password.

- User Search Pattern

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

- User Search Base (leave blank for top level)

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

- Search entire subtree

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

- Map LDAP Roles to Application Roles
Global Settings

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

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

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

- **Customize MySQL server name**

**Figure 15.6 MySQL Enterprise Monitor User Interface Settings: Customize MySQL server name**

These settings change the way host names are displayed in the MySQL Enterprise Monitor User Interface, typically by shortening the names to avoid cluttering the display with repetitive information:

- **The Show MySQL server names as** field controls whether the Monitor UI displays fully qualified domain names (the default); or only the host name, omitting the repetitive suffix such as .company_name.com; or host names transformed by a substitution expression, for example to turn a long multi-part host name into a short nickname.

The syntax for the substitution expression is a name-value pair separated by an equals sign, with a regular expression on the left side and the replacement text on the right side. The regular expression follows the Java syntax from http://download.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html. To keep special characters (particularly dot) from being interpreted within the regular expression, escape them with \ or make a single-item character class, [. ] for example. If the right side contains whitespace, a comma, or is an empty string, surround it with single or double quotation marks. You can use backreferences such as \$1, \$2, and so on to substitute parts of the regular expression into the replacement text; you cannot use the $ character in the replacement text except as part of a backreference. You can include more than one replacement expression by separating them with commas. Some examples include:
15.2 Manage Users

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

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

Figure 15.7 MySQL Enterprise Monitor User Interface: Manage Users

A user account is required to log into the MySQL Enterprise Monitor User Interface. There are four types of users with varying privileges:

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

- **dba role**
  Users with the dba role should be database administrators who are using MySQL Enterprise Monitor to monitor servers and queries. They have a similar set of privileges to those with the manager, except that they cannot do the following:
  - Cannot add/remove or change users.
  - Cannot change the My Oracle Support credentials used to communicate with the My Oracle Support website.
• Cannot change the HTTP proxy settings.
• Cannot change the Query Analyzer settings.
• Cannot change the LDAP authentication preferences.

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

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

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

• **read-only**

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

• **agent**

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

The Administrator defined during installation as having the root role is unique; this user cannot be deleted.

If you are logged in as an Administrator, you can add a new user by choosing the Manage Users link from the Settings tab. To create a user click the create user button, select a role for the user, and enter a user name and password.

When a new user first logs in, a dialog box opens requesting time zone and locale information. To change this information later, use the User Preferences tab. For more information, see Section 15.4, “User Preferences”.

If you installed the Advisors through the MySQL Enterprise Monitor User Interface, you have probably configured the settings for the root role user already. (See Section 15.1, “Global Settings” and following for more information about this topic.)

**Warning**

To receive support-related updates, configure the My Oracle Support (MOS) credentials for at least one user. These settings were set up on the first login to the Monitor UI. For information on changing these settings, see Section 15.1, “Global Settings”.

To edit an existing user's information, select the Manage Users link, then select the user to edit. Make your desired changes in the fields provided, then save the changes.

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

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

### 15.3 Logs
Logs

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

Use the Logs link to inspect, reset, and filter the various log files associated with the MySQL Enterprise Service Manager. For information about these log files, see Section D.1.1, “Log Files for the MySQL Enterprise Service Manager”.

The following image is an example of this screen.

Figure 15.8 MySQL Enterprise Monitor User Interface: Logs

The various categories of logs are shown in alphabetic order. The most recent changes to each log are shown in the Last Modified column. The number of entries in any specific log is shown under the Entries column.

To view detailed information, click the Log Name. This opens a separate browser window showing the date, time, alert type, and accompanying message.

On this screen, you can filter log information by the message type and by time period.

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

- All.
- Error.
- Warning.
- Information.
- Trace.
- Debug.
You can also adjust the number of items that appear on each page.

Press the clear all logs link to remove all log entries. To remove entries of a specific kind, click the clear logs link associated with the specific log. A confirmation dialog box lets you back out of this operation and avoid accidentally removing log information.

To clear log files of a specific age, see the Data Purge Behavior section of the Global Preferences page. For more information on this topic see Data Purge Behavior.

Use the edit log level link to change the type of error logged. The value selected from the Edit Log Level dialog box determines what appears under the Threshold column (second from the left in Data Purge Behavior).

Selecting Error from the list box creates the least number of log entries and Debug the most. Choosing None turns off logging altogether.

You can also download a compressed version of all the log files by downloading a Diagnostic Report via Settings.

### 15.4 User Preferences

**Note**

MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at [http://www.mysql.com/products/](http://www.mysql.com/products/).

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

**Figure 15.9 MySQL Enterprise Monitor User Interface: User Preferences**

To change your password, enter 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 can also adjust your time zone and locale information from this tab. The settings on this tab apply only to the user who is currently logged in.

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

15.5 Diagnostics Report

This chapter describes the Diagnostics Report.

Introduction

You can obtain a diagnostics report file for either individual servers, or entire server groups. To get a
diagnostic report file, select **Diagnostics Report** from the **Settings** menu. The information is provided
as a time stamped Zip file (such as `support-20130815T2238.zip`) that is downloaded to the
machine. The information contained in the report includes detailed information about your server (or
multiple servers if you selected a server group), including configuration, hardware, MySQL options/
variables and historical graphs. To view the information extracted, unzip the downloaded file and
double-click the **index.html**.

This report is especially useful for debugging the MySQL Enterprise Service Manager and the MySQL
Enterprise Monitor Agent. When filling out a My Oracle Support (MOS) ticket, it is common practice for
the support team to ask for this report.

Diagnostics Report File Contents

- **audit.log**: The Audit log file.
- **catalina.out**: A Tomcat log file.
- **java-threads.dot**: A list of the current Java threads and the dependencies.
- **java.props**: The current Java configuration properties.
- **java.threads**: A list of the current Java threads and their backtrace.
- **mysql-monitor.log**: The general MySQL Enterprise Service Manager log file.
- **mysql-monitor-full.log**: The full MySQL Enterprise Service Manager log file, that also
contains stack traces.
- **mysql-monitor-agent-full.log**: A full built-in MySQL Enterprise Monitor Agent log file, that
also contains stack traces.
- **preferences.properties**: The MySQL Enterprise Monitor preference settings.
- **product usage.html**: A usage report for each MySQL Enterprise Monitor User Interface page.
- **query.instanceOverview.html**: An HTML list of the current query instance related information.

Note
The format of this file changed in 3.0. It is now listed as one Asset per block,
instead of having one row per Asset inventory item.

- **Replication 1.dot**: The calculated MySQL server replication structure.
- **root.csv**: A copy of your main MySQL Enterprise Monitor log file.
Diagnostics Report File Contents

- **server.props**: A copy of your server properties.
- **tomcat.log**: A Tomcat log file.
Chapter 16 Reports and Graphs

Table of Contents

16.1 All Timeseries Graphs .................................................................................................................. 161
16.2 InnoDB Buffer Pool Usage .......................................................................................................... 162

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

16.1 All Timeseries Graphs

Navigate to Reports & Graphs, and choose All Timeseries Graphs.

Note
While the main Overview page only shows connection related graphs for your MySQL instances, the All Timeseries Graphs page lists all graphs that relate to time.

The data shown in the graphs is determined by the server or group of servers selected in the server tree. Each graph is only displayed if there is appropriate data for the selected servers; for example, only slave servers have a Replication Slave Delay graph, and the Cluster graphs only apply to MySQL Cluster servers.

Graph Types

There are five main types of graphs:

• Individual: A single Asset has multiple data sets graphed on a chart. For example, counts of SELECT, INSERT, UPDATE, and DELETE statements on a single Instance.

• Combined: Multiple assets have a single data set, each graphed on one chart. For example, the count of selects for each of the five MySQL instances of a group.

• Breakout: One (smaller) graph per Asset in a collection, showing one or more data sets on each individual graph. For example, one graph per CPU on a Host, or in a cluster.

• Aggregate: One graph per collection of Assets, where the data sets across all Assets are combined via an aggregation operator. For example, one graph with each of the SUM(SELECT), SUM(INSERT), SUM(UPDATE), and SUM(DELETE) across the collection.

• Treemap: A 2D hierarchical proportional-representation graph. See Section 16.2, “InnoDB Buffer Pool Usage” for an example.

Graph Behavior

Other behavioral notes about the graphs:

• The graphs are context aware, and they display the most appropriate data for the Group or specific type of Asset (such as a MySQL Instance) that is selected. For example, when a Group is selected, details about the MySQL Instances in that Group will be presented in Aggregate. When a specific MySQL Instance is selected, its details will be presented directly. When a Host is selected, information about its File Systems will be presented in Aggregate, but when a specific File System is selected, it will be graphed individually.
• The graphs show "item time spans" in large time ranges, and each data point represents a small time range. Because each graph is limited by the number of pixels per width, each data point must fit into the available pixels in the graph, and as such, the small time range adjusts to the graph size and range.

For example, as a graph spans a larger segment of time, each data point also covers a larger segment of time. A data point could cover 3 minutes on a single day graph, or 6 minutes on a two-day graph, which affects the height and range. In this example, a data point could read as 3,000 on the first graph, and 6,000 on the second.

• Hovering over a series item will isolate (focus on) the particular series. This might show a specific timestamp, or display a range of time where each data point represents multiple units.

• Clicking a series item will remove it from the graph, and the X-axis and Y-axis values will adjust themselves accordingly.

• Selecting a range (via mouse dragging) yields Query Analysis (QUAN) option for the particular range.

• The Graph Height slider adjusts the height and size of every graph.

• Graphs may be exported as CSV data, and may be viewed as either stacked or line graphs.

• Group based graphs include a "Combined" graph type, which shows a series for every asset within that group (per host for host specific graphs, or per MySQL instance for MySQL specific graphs), rather than just the avg/min/max of all servers within the group. An example use case is for determining the cause of a spike.

• Graphs can be reordered and filtered. Reordering is preserved if the page is refreshed.

• Under some circumstances, a graph may contain a gap, which could be caused by several different conditions:

  • After the Service Monitor was stopped, Agent specific metrics (such as the Database Activity Graph) can "expect" a 1 or 2 minute gap at the beginning of the offline period, and at the time the Service Monitor was started.

  • Some graphs, such as the CPU Utilization Graph, might have gaps between the time the Service Monitor is stopped and started if the Agent is running on the same host as the Service Monitor.

  • When it is impossible to evaluate data that is misaligned, so a gap is generated rather than misrepresenting the data.

All graphs support the ability to drag and select a specific area of the graph. When you select an area of the graph, the display adds a link to the Query Analyzer page to display the queries that were being executed during the selected time period. You can also export data for a particular graph by clicking Export as CSV while hovering over the graph you wish to export, or Export as PNG for a PNG image.

16.2 InnoDB Buffer Pool Usage

The InnoDB Buffer Pool Usage Report displays the amount of space used in the InnoDB buffer pool and how the space is used. The report is displayed in grid format. Each block in the grid represents a particular type of data stored in the buffer pool. Click a block to display more details.

For more information on the InnoDB Buffer Pool, see InnoDB Buffer Pool Configuration and The InnoDB Buffer Pool

Important

This report requires the INFORMATION_SCHEMA.INNODB_BUFFER_PAGE table, which is available in MySQL Server version 5.5.28 or higher.
Running the InnoDB Buffer Pool Usage Report

To run the usage report, do the following:

1. Navigate to the Reports & Graphs drop-down menu.

2. Select InnoDB Buffer Pool Usage.
   
The Generate Report page is displayed.
   
   This page displays a warning about the table and resource requirements of the report generation process and prompts you to select a MySQL Server to run the report against.

   [Important]
   
The report can take some time to return results. If no data is returned within 2 minutes, the report times out and an error is displayed.

3. Select the MySQL server from the asset tree.
   
The Generate Report page is displayed.

4. Click Generate Report.
   
The Loading buffer pool report progress message is displayed.

   [Note]
   
   If you click Reload while the report is generating, the report generation process is cancelled and restarted. If you navigate away from the progress page, the report generation process is cancelled.

5. The report is displayed.
Chapter 17 Events

Table of Contents

17.1 Closing an Event ................................................................. 167
17.2 Notification of Events .......................................................... 168

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

Once an advisor is enabled, it runs at set intervals. If it finds nothing of interest, it does not create events, which also means that notifications are not generated.

Events are defined using a number of different levels according to the severity of the alert. The eight levels are:

• **Failure**: Indicates a failure for the given rule or event. Usually, this indicates that the rule on which the event is based has failed, probably due to a typo or other error. Occasionally, it can indicate a failure to retrieve the required information for a given rule.

• **Emergency**: The event is an emergency and requires immediate attention. This is a panic condition, and means the system is unusable.

• **Critical**: The event is critical and requires immediate attention. Critical events normally indicate that a serious problem has occurred or is about to occur. Examine the event, determine the cause, and fix the issue as soon as possible.

• **Warning**: The event is something to be aware of, but does not affect the operation of your server.

• **Notice**: The event is a notice for informational purposes. Notice events call attention to issues that do not affect the operation of your server, such as a minor configuration issue.

Note
This was formerly known as the **Informational** level.

• **Success**: The rule executed successfully with no issues. The result of the rule is OK. It also signals when an event that was previously in a Critical or Failure state returns to normal.

• **Unknown**: The current status of the event/rule cannot be determined.

• **Closed**: The issue has been corrected and marked closed.

For convenience, the event states are also represented by icons, as defined in the following table. The table also shows the relative level of the alert from the highest (Emergency) to the lowest (Unknown). The order represented here is used when identifying when an event has transitioned between levels (for example, from Success to Critical), hence triggering a notification, and when sorting events based on their current state.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔥</td>
<td>A red and orange flames icon indicates an emergency alert.</td>
</tr>
<tr>
<td>⚠️</td>
<td>An round red icon indicates a critical alert.</td>
</tr>
</tbody>
</table>
## Icon Description

- **A triangular yellow icon indicates a warning.**
- **An blue octagon with the letter "i" indicates an informational alert.**
- **A green check beside an event indicates that the Advisor has run successfully and no alert created.**
- **A skull icon indicates that the status of the Advisor is unknown.**

All Events are displayed on the **Events** page. Emergency and Critical events are also displayed on the **Overview** dashboard. The notification group or groups associated with a specific advisor receive a notification when an alert is triggered. For more information about creating notification groups, see **Chapter 18, Event Handling.**

To view open events, click the **Events** tab. The tree-view on the left determines which server or server group these events belong to. Open events are shown in tabular format.

**Figure 17.1 MySQL Enterprise Monitor User Interface: Events Screen with Search enabled**

The event table has the following columns:

- **Current:** An icon indicating the current severity of the alert.
- **Worst:** An icon indicating the worst severity recorded for this alert.
- **Subject:** The name of the asset the alert applies to, which is typically a MySQL instance.
- **Topic:** A short description of the rule that is violated. Click the + on the left of the event entry to view additional information.
- **Time:** The approximate time the event occurred.
- **Actions:** Possible actions, where clicking X will open the **Close** dialog menu.
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:

- Time Range - this time is filtered by the time the event was first seen
- State
- Current Status
- Worst Status
- Advisors

Choose the options and click the Fillter button to refresh the display. To limit the number of items that appear on a page, choose a different value from the Show entries drop-down list box.

The drop-down list box showing severity status has the options: Any, All Alerts, Emergency, Critical, >=Critical, Warning, >=WarningNotice, Warning, Success, and Unknown. The default option (any) shows all events, and also those advisors that have run successfully.

A 'successful' Event is one that currently evaluates past the level of the "Success" threshold, and is indicated by a green tick icon.

The All Alerts shows only those advisors that have been violated.

Some columns can be sorted by clicking the individual column headings. The alerts shown in Figure 17.1, "MySQL Enterprise Monitor User Interface: Events Screen with Search enabled".

Note

By default, the columns are sorted by Current and Time. If the default sorting is changed, the changes are stored in the browser’s cache. If the browser’s cache is cleared, the default sort is restored.

The server shown in Figure 17.1, "MySQL Enterprise Monitor User Interface: Events Screen with Search enabled", is filtered for Any. Typically, when filtering by severity you choose All Alerts and, if you see an Emergency Critical, Warning, or Notice alert, use the All Alerts filter to see when the rule last ran successfully. Narrowing down the time frame can help determine the cause of the alert.

Besides filtering for severity, you can filter for a specific time period using the Time Range dropdown. You can also filter by specific rules or categories of rules. The State dropdown list box lets you choose Open or Closed events. To avoid excessive scrolling, you can also limit the number of events shown on a specific page.

For more information about an alert, click anywhere on the row with the rule name. A slider window appears, showing a description of the alert and the exact time of occurrence. This slider provides links to useful resources and advice for resolution.

For expression-based advisors, where possible, the results of the expression are listed in the Evaluated Expression section of the Details frame.

17.1 Closing an Event

Advisor’s generate events when the threshold defined on the Advisor is breached. Investigate the issue that triggered the event; rectify the issue or problem (or choose to ignore it); then close the event when you are satisfied it does not have a significant impact on your servers.

Some of the advisors identify transient or temporary issues, such as a replication slave being unavailable. For these advisors, you can schedule events to automatically be closed when the event moves from notification status back to the Ok state.
When auto-close is enabled, the event remains open while the condition that triggered the event is still in effect. When the condition is no longer in effect, the event is automatically closed. You can also manually close such events before resolving the issue. Events can also be closed by event handlers. For more information on handling events, see Chapter 18, Event Handling.

**Important**

Not all Advisors generate events which can be auto-closed. See Chapter 20, Expression-Based Advisor Reference and Chapter 21, GUI-Based Advisor Reference for more information on the Advisors which support auto-close.

Events which support auto-close are closed by the Default Auto-close Policy after the event which triggered them is no longer in effect. For more information on Default Auto-close Policy, see Default Auto-close Policy. It is possible to override the Default Auto-close Policy by setting Auto-Close Events to No in an Event Handler.

To close an individual event, click the [X] icon in the Actions column. Document the resolution using the Notes text area and choose the Close Events button. During the closing operation, you can also reconfigure the rule scheduling that triggered this event by checking the checkbox titled "After closing, take me to the page for adjusting schedules of Advisor(s) that reported these events." This option will open the Advisors configuration page with the related Advisors selected.

For more information on configuring advisor scheduling and auto closing, see Table 19.3, "Advisor Edit Menu Controls".

To close a number of alerts simultaneously, select the checkbox beside each event to close and click the Close Selected Events button.

When closing individual or multiple events, a notification window indicates what operations have been completed. The events remain in the displayed event list, but the close link is instead a link to the resolution notes. You can update the active list by clicking filter to re-filter the event display.

A historical list of all events, including closed events, is available by setting the Current Severity to Closed. The list shows all of the closed events for a given time range and the servers selected in the server tree. Historical data is limited by the purge data settings.

**Automatic Closing of Events**

If a custom advisor is deleted, or one of the default advisors is made redundant and removed as part of an upgrade, their events can be orphaned. The system automatically closes events which have no advisor linked to them. A note is added to the event stating why it was closed.

Auto-closed events send a notification only if notifications were sent for any previous state transitions. If no other notifications were sent, no notification is sent for the auto-close.

**17.2 Notification of Events**

By default, notification of an event takes place when an Advisor is executed and the severity level for a given Advisor changes from a lower status to a higher one. This occurs on the first change to each individual severity level until the event is closed.

This behavior is set when creating an event handler (Configuration, Event Handling, Create Event Handler). The SMTP Notification Policy and SNMP Notification Policy options include:

- **Notify on event escalation**: Notify when the status changes from a lower status to a higher one, such as Critical to Emergency.
- **Notify on any status change**: Notify when the status changes, either high to low or low to high.
• **Always notify**: Does not take into account escalation, and always sends a notification if one of the defined Event Statuses is triggered.

If auto-close is enabled for a given advisor, this notification sequence is still valid, but the event is automatically closed when the event reaches the ok state. This re-enables notification when the advisor creates a new event.
Chapter 18 Event Handling

Table of Contents

18.1 Event Handling Page .................................................. 171
18.1.1 Event Handlers List .................................................. 171
18.1.2 Email Notification Group Controls ............................... 172
18.1.3 Email Settings ....................................................... 174
18.1.4 Email Notification Status ........................................ 174
18.1.5 SNMP Settings ...................................................... 174
18.1.6 SNMP Notification Status ...................................... 175
18.2 Creating Event Handlers .............................................. 175
18.2.1 Event Action Log ................................................... 177
18.2.2 Suspending an Event Handler ................................. 178

Event handlers define who is notified, and how they are notified, when the thresholds on Advisors are breached and how the event is treated after the status changes.

18.1 Event Handling Page

This section describes the Event Handling page of MySQL Enterprise Service Manager.

To display the Event Handling page, select Event Handling from the Configuration drop-down menu.

The Event Handler page is grouped in the following sections:

- **Event Handlers**: Lists the event handlers defined on the system. The Default Auto Close Policy is present by default and cannot be edited.
- **Email Notification Groups**: lists the email notification groups defined on the system.
- **Email Settings**: enables you to define the email configuration, such as SMTP server, username and password to use for all outgoing emails.
- **Email Notification Status**: displays the success or failure of the last email sent.
- **SNMP Settings**: enables you to define the SNMP trap configuration, such as SNMP version, SNMP targets, and so on.
- **SNMP Notification Status**: displays the success or failure of the last SNMP trap sent.

18.1.1 Event Handlers List

The Event Handlers section lists all event handlers defined on the system.

Figure 18.1 Event Handlers section

Event Handlers section contains the following controls:

Table 18.1 Event Handler List Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Event Handler</td>
<td>Opens the Create Event Handler dialog. For more information, see</td>
</tr>
<tr>
<td>Show * Entries</td>
<td>Select the maximum number of event handlers to display.</td>
</tr>
</tbody>
</table>
Email Notification Group Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handler Name</td>
<td>Lists the names of the event handlers.</td>
</tr>
<tr>
<td>State</td>
<td>Lists the state of the event handler. Possible states are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Active</strong>: the event handler is running.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Suspended</strong>: the event handler is not running.</td>
</tr>
<tr>
<td>Groups</td>
<td>Lists the groups assigned to the event handler.</td>
</tr>
<tr>
<td>Assets</td>
<td>Lists the assets assigned to the event handler.</td>
</tr>
<tr>
<td>Advisors</td>
<td>Lists the Advisors assigned to the event handler.</td>
</tr>
<tr>
<td>Statuses</td>
<td>Lists the statuses assigned to the event handler.</td>
</tr>
<tr>
<td>Actions</td>
<td>Lists the SMTP or SNMP actions assigned to the event handler.</td>
</tr>
<tr>
<td>Search</td>
<td>Enables you to search for specific event handlers.</td>
</tr>
</tbody>
</table>

**Default Auto-close Policy**

The **Default Auto-close Policy** closes events after they change status. If a threshold is defined for an advisor, and the threshold is breached, an event is displayed in the **Events** page. If it changes status to a lower priority status, or to a status without a defined threshold, the default auto-close policy closes the event.

| Note | The **Default Auto-close Policy** event handler is the only event handler created by default. |

This policy does not apply to all Advisors. Some Advisors, such as **MySQL Server Has Been Restarted**, are too important to auto-close.

For more information on the Advisors, and how the default auto-close applies, see Chapter 20, *Expression-Based Advisor Reference* and Chapter 21, *GUI-Based Advisor Reference*.

| Important | It is not possible to edit this Event Handler, but it is possible to override it using the Auto-Close Events option in the Create Event Handler dialog. |

**18.1.2 Email Notification Group Controls**

This section describes the controls on the **Email Notification Group** section.

**Figure 18.2 Email Notification Groups section**

The **Email Notification Groups** contains the following controls:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Notification Group button</td>
<td>Opens the Create Notification Group dialog. For more information, see</td>
</tr>
<tr>
<td>Group Name</td>
<td>Lists the names of the notification groups.</td>
</tr>
<tr>
<td>Recipients</td>
<td>Lists the recipients’ email addresses.</td>
</tr>
</tbody>
</table>
Email Notification Group Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject Line</td>
<td>The subject line of the notification emails.</td>
</tr>
<tr>
<td>SMS</td>
<td>Status of SMS encoding. The following values are possible:</td>
</tr>
<tr>
<td></td>
<td>• <strong>true</strong>: SMS encoding is enabled.</td>
</tr>
<tr>
<td></td>
<td>• <strong>false</strong>: SMS encoding is not enabled.</td>
</tr>
<tr>
<td>MEM Admin</td>
<td>Status of emails regarding MySQL Enterprise Monitor. The following values</td>
</tr>
<tr>
<td></td>
<td>are possible:</td>
</tr>
<tr>
<td></td>
<td>• <strong>true</strong>: critical MySQL Enterprise Monitor emails will be sent to this</td>
</tr>
<tr>
<td></td>
<td>notification group.</td>
</tr>
<tr>
<td></td>
<td>• <strong>false</strong>: no email related to MySQL Enterprise Monitor will be sent to</td>
</tr>
<tr>
<td></td>
<td>this notification group.</td>
</tr>
</tbody>
</table>

Creating an Email Notification Group

You can define email notification groups using the **Create Group** dialog. To open the **Create Group** dialog, click **Create Notification Group** in the **Email Notification Groups** section of the Event Handling page.

Figure 18.3 Create Group Dialog

To create a notification group, do the following:

1. On the Event Handlers page, select **Create Notification Group**.
   
   The **Create Group** dialog is displayed.

2. In the **Group Name** field, specify a group name to uniquely identify this notification group.

3. In the **Recipients** field, add a comma-separated list of email addresses. These are the addresses to which the notifications will be sent.

4. In the **Subject Line** field, specify the subject line which will be added to every email sent by this notification group.

5. If required, select **SMS (Use SMS encoding for this notification group)**.

6. If you want to send information regarding the status of MySQL Enterprise Monitor to the recipients of this notification, select the **MEM Admin** checkbox. Only critical system messages will be included.

7. Click **Save Notification Group**.

   The notification group is available for use in event handlers.
18.1.3 Email Settings

The Email Settings section enables you to define the email configuration, such as SMTP server, username and password to use for all outgoing emails.

Figure 18.4 Email Settings section

The Email Settings pane contains the following controls:

Table 18.3 Email Settings Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Email Notifications</td>
<td>Select to activate the email settings controls.</td>
</tr>
<tr>
<td>From Address</td>
<td>The email address added to the From field of all emails sent from MySQL Enterprise Monitor.</td>
</tr>
<tr>
<td>SMTP server</td>
<td>The outgoing email server address</td>
</tr>
<tr>
<td>SMTP Server Login</td>
<td>The username for the SMTP server</td>
</tr>
<tr>
<td>Update Password on Save</td>
<td>Select to activate the password fields.</td>
</tr>
<tr>
<td>Disable JavaMail TLS/SSL</td>
<td>Select if the SMTP server does not require an encrypted connection.</td>
</tr>
<tr>
<td>On Save, Send Test Email Message To</td>
<td>Enter an email address if you want to send a test email when the changes are saved.</td>
</tr>
<tr>
<td>Save Email Settings</td>
<td>Saves the Email Settings and sends a test email if an address is defined in the On Save, Send Test Email Message To field.</td>
</tr>
</tbody>
</table>

18.1.4 Email Notification Status

The Email Notification Status section displays the success or failure of the last email sent, and an error message describing why the sending failed.

18.1.5 SNMP Settings

The SNMP Settings section enables you to define the SNMP trap configuration, such as SNMP version, SNMP targets, and so on.

Figure 18.5 SNMP Settings section
The SNMP Settings pane contains the following controls:

### Table 18.4 SNMP Settings Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable SNMP Notifications</td>
<td>Activates the SNMP configuration fields.</td>
</tr>
<tr>
<td>Use SNMP v1/v2</td>
<td>Choose the version of SNMP you intend to use.</td>
</tr>
<tr>
<td>Target and Port Number</td>
<td>IP address and Port number of the system which will receive the SNMP Traps.</td>
</tr>
<tr>
<td>Community String</td>
<td>SNMP community string. Default value is <code>public</code>.</td>
</tr>
<tr>
<td>Use the remote MySQL agent host IP address as the SNMP</td>
<td>Defines the source IP address included in the trap.</td>
</tr>
<tr>
<td>trap agent address for Advisor traps (optional)</td>
<td>• Disabled: the trap uses the IP address of the service manager.</td>
</tr>
<tr>
<td></td>
<td>• Enabled: the trap uses the IP address of the agent monitoring the host</td>
</tr>
<tr>
<td></td>
<td>for which the advisor was triggered.</td>
</tr>
<tr>
<td>SNMP trap agent address for internally generated traps</td>
<td>Defines the source IP address included in traps generated by MySQL</td>
</tr>
<tr>
<td>(optional)</td>
<td>Enterprise Service Manager</td>
</tr>
<tr>
<td>On Save send test trap</td>
<td>Send a test trap message when Save is clicked. Select one, or more, of the</td>
</tr>
<tr>
<td></td>
<td>trap types from the list. One trap is sent for each option selected.</td>
</tr>
</tbody>
</table>

### 18.1.6 SNMP Notification Status

The SNMP Notification Status section displays the success or failure of the last trap sent, and an error message describing why the sending failed.

### 18.2 Creating Event Handlers

Event handlers enable you to create a condition which, when met, triggers notifications to concerned parties such as DBAs, System Administrators and so on.

The following condition criteria can be defined:

- **Assets and Groups**: enables you to select multiple assets or multiple groups to monitor.

  **Important**
  
  It is possible to define both Assets and Groups in an event handler, but is not recommended. It is recommended that you create the event handler using either Assets or Groups, not both. If you define Assets and Groups in an event handler, notifications will only be sent for the defined Assets which also exist in the defined Groups.

- **Advisors**: enables you to select multiple Advisors to evaluate.

- **Event Statuses**: enables you to select multiple statuses to monitor.

To create an event handler, click **Create Event Handler** in the Event Handlers section on the Event Handlers page.

The Create Event Handler dialog is displayed:
Creating Event Handlers

Figure 18.6 Create Event Handler Dialog

Table 18.5 Create Event Handler Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Handler Name</td>
<td>Specify a name which uniquely identifies the new event handler.</td>
</tr>
<tr>
<td>Assets</td>
<td>Select the individual assets to monitor from the Assets drop-down list. If this field is left blank, all assets are included in the event handler's condition, unless one or more groups are defined. If groups are defined, and the asset field is blank, the event handler's condition includes groups only. The Assets drop-down list displays the Assets in their groups, if groups are defined. If no groups are defined, it lists the assets. It is not possible to select groups in the Assets field. You must expand the group to select individual assets.</td>
</tr>
<tr>
<td>Groups</td>
<td>Select the groups of assets to monitor. If this field is left blank, all groups are included in the event handler's condition, unless one or more assets are defined. If assets are defined, and the group field is blank, the event handler's condition includes assets only.</td>
</tr>
<tr>
<td>Advisors</td>
<td>Select the Advisors. If this field is left blank, all advisors are included in the event handler's condition.</td>
</tr>
<tr>
<td>Event Statuses</td>
<td>Select the statuses for which you want to receive notifications.</td>
</tr>
</tbody>
</table>
### Event Action Log

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTP Notification Groups</td>
<td>Select the groups you want to notify.</td>
</tr>
<tr>
<td>SMTP/SNMP Notification Policy</td>
<td>Select one of the following:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Notify on event escalation</strong>: sends a notification only if the event changes to a higher priority. From Warning to Critical, for example. No notification is sent if the event changes to a lower priority. From Critical to Warning, for example.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Notify on any status change</strong>: sends a notification if the status changes to any other status.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Always notify</strong>: sends a notification every time the defined event status(es) are broken. For example, if Critical and Emergency are defined in the Event Status field, and Always notify is selected, a notification will be sent every time the Critical and Emergency events are triggered for the selected Advisors.</td>
</tr>
<tr>
<td>Note</td>
<td>This setting can generate a very high volume of notifications but can be useful if you replicate event states to a secondary system, such as a ticketing system.</td>
</tr>
<tr>
<td>SMTP Rate Limit</td>
<td>Defines the maximum number of SMTP notifications which can be sent per minute. The default is 40.</td>
</tr>
<tr>
<td></td>
<td>If this rate is exceeded, no further notifications will be sent until the period ends and the new period begins. An event is raised listing the event handler whose rate limit was exceeded and the rate defined on that event handler.</td>
</tr>
<tr>
<td></td>
<td>These events are not auto-closed and are updated only for the first failure.</td>
</tr>
<tr>
<td>Send SNMP Traps</td>
<td>Defines whether SNMP Traps are used for notifications.</td>
</tr>
<tr>
<td>Auto-Close Events</td>
<td>Defines whether the events are closed after the trigger status changes. The following values are possible:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Yes</strong>: the default auto-close policy is honored and the events are closed when the conditions defined are no longer met.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No</strong>: the default auto-close policy is ignored and the events remain open in the Events page even after the conditions are no longer met.</td>
</tr>
</tbody>
</table>

**Important**

If you leave the Assets, Groups, and Advisors fields empty, the event handler can generate an extremely high volume of emails, depending on the number of assets monitored. It is recommended to create event handlers which address specific requirements and contain strictly defined criteria.

**18.2.1 Event Action Log**

If an event handler is triggered, the action taken is displayed in the Action Log \[n\] section at the bottom of the expanded event, where \[n\] represents the number of actions logged for that event.

The Action Log \[n\] lists the time the action was taken, the type of action (SMTP or SNMP), the failure or success of the action, and the triggering policy used to trigger the event. The **Triggering Policy** column lists the names of the event handlers which triggered the actions.
18.2.2 Suspending an Event Handler

To stop an event handler, select **Suspend Event Handler** from the event handler's drop-down menu. A prompt is displayed enabling you to enter notes on why the event handler was suspended, and confirm the suspension.
Chapter 19 Advisors

Table of Contents

19.1 Advisors Page .......................................................... 179
19.2 Advisor Types .......................................................... 183
19.3 Advisor Thresholds ...................................................... 184
19.4 Advisor Schedules ....................................................... 185

This chapter describes MySQL Enterprise Advisors.

Advisors filter and evaluate the information collected by the Monitoring Agents and present it to the Events page when defined thresholds are breached. There are more than 200 Advisors, all of which are enabled by default.

The following topics are described in this chapter:

• Section 19.1, “Advisors Page”
• Section 19.2, “Advisor Types”
• Section 19.3, “Advisor Thresholds”
• Section 19.4, “Advisor Schedules”

19.1 Advisors Page

This section describes the main Advisors page.

To display the Advisors page, select Advisors from the Configuration drop-down menu.

Figure 19.1 Advisors Page

The components and controls of the Advisors page are as follows:

Table 19.1 Advisor Page Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Selected Advisors</td>
<td>Opens the edit dialog for the selected advisor. This control can also be used for multiple Advisors, but it is only possible to change the Schedule</td>
</tr>
</tbody>
</table>
Advisor Categories

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable Selected Advisors</td>
<td>Disables all selected Advisors.</td>
</tr>
<tr>
<td>Create Advisor</td>
<td>Opens the Create Advisor page.</td>
</tr>
<tr>
<td>Import/Export</td>
<td>Opens the Custom Rule/Graph Export page. Note: This functionality is for</td>
</tr>
<tr>
<td></td>
<td>custom rules and graphs only.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all Advisors.</td>
</tr>
<tr>
<td>Expand All</td>
<td>Expands all categories.</td>
</tr>
<tr>
<td>Collapse All</td>
<td>Collapses all categories and clears all selections.</td>
</tr>
<tr>
<td>Filter Advisors</td>
<td>Expands or collapses the Advisor filter. The Advisor filter enables you to</td>
</tr>
<tr>
<td></td>
<td>filter the Advisors, groups and assets.</td>
</tr>
</tbody>
</table>

Advisor Categories

The following types of Advisor are provided:

- **Administration**: Checks the MySQL instance installation and configuration.
- **Agent**: Checks the status of each MySQL Enterprise Monitor Agent.
- **Availability**: Checks the availability of the MySQL process and the connection load.
- **Backup**: Checks whether backup jobs succeed or fail, required resources, and information about MySQL Enterprise Backup specific tasks.
- **Cluster**: Checks the status of the monitored MySQL Cluster.
- **Graphing**: Data for graphs.
- **Memory Usage**: Indicate how efficiently you are using various memory caches, such as the InnoDB buffer pool, MyISAM key cache, query cache, table cache, and thread cache.
- **Monitoring and Support Services**: Advisors related to the MySQL Enterprise Monitoring services itself.
- **Operating System**: Checks the Host Operating System performance.
- **Performance**: Identifies potential performance bottlenecks, and suggests optimizations.
- **Query Analysis**: Advisors related to Queries and Query Analysis.
- **Replication**: Identifies replication bottlenecks, and suggests replication design improvements.
- **Schema**: Identifies schema changes.
- **Security**: Checks MySQL Servers for known security issues.

It is also possible to create custom Advisors.

To display the Advisors in each category, click on the Category name. For a full description of the default advisors, see Chapter 21, *GUI-Based Advisor Reference* and Chapter 20, *Expression-Based Advisor Reference*.

Advisors configure the type of data collected by the Agent. If you do not want to monitor for a specific type of data, disabling the Advisor responsible for that data type instructs the Agents to stop collecting that data.
Advisor Listing Table

The listing table displays all categories, Advisors, monitored instances, and displays information on the configuration of the Advisors.

The configuration information is displayed in the following columns:

**Table 19.2 Advisor Information Listing**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Displays the Advisor name, group name, and monitored instance name. To expand the Advisor, click the expand icon.</td>
</tr>
<tr>
<td>Info</td>
<td>Click to display a tooltip which describes the Advisor.</td>
</tr>
<tr>
<td>Coverage</td>
<td>Displays the Advisor's coverage of the monitored instance. If the Advisor has been edited for a specific instance, this field is empty for that instance. If the default Advisor settings are used, this field displays <em>(Covered)</em>.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Displays the defined evaluation schedule. If the Advisor is disabled, this field displays <em>Disabled</em> for the level at which it was disabled, Advisor, Group or monitored instance.</td>
</tr>
<tr>
<td>Event Handling</td>
<td>Displays the event handling status icons. For more information, see Chapter 18, Event Handling.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Displays the Advisor's configuration details, thresholds, and so on.</td>
</tr>
</tbody>
</table>

Advisor Menu

To open the Advisor menu, click the drop-down icon next to the Advisor's name.

**Figure 19.2 Advisor Menu Control**

The Advisor menu is displayed:

**Figure 19.3 Advisor Popup Menu**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Advisor</td>
<td>Opens the Create New Advisor page and appends – Copy 1 to the Advisor name. This enables you to define new Advisors based on existing ones.</td>
</tr>
<tr>
<td>Edit Advisor Configuration</td>
<td>Opens the Edit Advisor dialog. This enables you to change the parameters and schedule of the selected advisor.</td>
</tr>
</tbody>
</table>

**Note**

This option is only available for expression-based Advisors.
Group and Host Menu

Each advisor contains the list of all groups defined in MySQL Enterprise Monitor. To see these groups, expand the contents of the Advisor by clicking on the Advisor’s name. This enables you to specify the Advisors you want to run for each group.

The top-level advisor contains the global configuration for all groups. That is, the configuration at the advisor-level applies to all groups and hosts it contains. Each nested group, and the monitored hosts contained in the group, have a drop-down menu enabling you to override the global configuration for each group or host, or disable the advisor for the specific group or host. Any change in advisor configuration at the group or host level, overrides the global configuration specified at the advisor level.

To open the Group menu, expand the Advisor and select the drop-down icon next to the Group name. The menu contains the following items:

- **Override Advisor Configuration**: opens the Advisor edit dialog, enabling you to change the Advisor’s configuration for the assets in the group. Changes made at the group level, only affect the assets within the group.

  **Important**
  If a host, Host1 for example, exists in multiple groups and a configuration override is applied to one of those groups, it does not affect Host1. Data is still collected and events generated for Host1 because it exists in different groups within the same advisor. To ensure the override applies to Host1, you must apply the same override to Host1 in each group which contains it.

- **Disable Advisor**: disables the Advisor for the selected group or host.

Advisor Filter

The Advisor filter enables you to search for specific Advisors, groups, hosts, or assets using text or regular expressions. To open the filter, click the filter button.

The Advisor Filter is displayed:

**Figure 19.4 Advisor Popup Menu**

**Table 19.4 Advisor Filter Controls**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advisors</td>
<td>Opens a drop-down menu listing all available Advisors. You can select multiple Advisors.</td>
</tr>
</tbody>
</table>
### Advisor Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groups</td>
<td>Opens a drop-down menu listing all defined groups. You can select multiple groups.</td>
</tr>
</tbody>
</table>
| Asset Name    | Opens a drop-down menu listing the available search types:  
|               |     • Contains  
|               |     • Doesn't Contain  
|               |     • Regex  
|               |     • Negative Regex                                                                                                                                 |
| Value         | Free text field for the search term.                                                                                                                                                             |
| Filter        | Filters the Advisors list based on the search terms.                                                                                                                                               |
| Filter & Expand| Filters the Advisors list based on the search terms and expands the categories and Advisors to display the search results.                                                                  |
| Reset         | Resets all filter values.                                                                                                                                                                          |

### 19.2 Advisor Types

There are two types of Advisor:

- Expression-based
- GUI-based

#### Expression-based Advisors

The majority of Advisors use a simple expression to evaluate the data collected by the monitoring Agent. These expressions use the following syntax:

```
%VariableName% operator THRESHOLD
```

where:

- `%VariableName%` is the monitored value. The variables correspond to elements of the data collected by the Agent.
- `operator` is a mathematical operator such as `<`, `>`, `!`, `=`, and so on.
- `THRESHOLD` is the Advisor-defined limit for the monitored value.

These expression-based Advisors evaluate the monitored values against the defined thresholds. Expression-based Advisors can evaluate percentage values, time/duration values, or check for the existence of specific configuration values.

More complex expressions are also used by concatenating a variety of different variables. It is also possible to perform calculations on the results returned by these variables within the expressions.

#### GUI-based Advisors

The GUI-based Advisors contain more configuration options than the expression-based Advisors. These Advisors evaluate many more values than the expression-based Advisors and do not use the same expression-based evaluation system.

The following example shows the General section of the Agent Health Advisor:
19.3 Advisor Thresholds

Thresholds are the predefined limits for Advisors. If the monitored value breaches the defined threshold, an event is generated and displayed on the Events page for the asset.

Advisor thresholds use a variety of different value types, depending on the monitored value. Some use percentages, such as percentage of maximum number of connections. Others use timed durations, such as the average statement execution time. It is also possible to check if specific configuration elements are present or correct.

The following thresholds, listed in order of severity, can be defined for most Advisors:

- **Notice**: issues which do not affect the performance of the server, but can be used to indicate minor configuration problems.

- **Warning**: issues which do not affect the performance of the server, but may indicate a problem and require investigation.

- **Critical**: indicates a serious issue which is affecting or will affect the performance of the server. Such issues require immediate attention.

- **Emergency**: indicates a serious problem with the server. The server is unusable or unresponsive and requires immediate attention.

**Note**

Not all Advisors require threshold parameters, others do not have any parameters, such as the **Graphing** Advisors.

The following image shows an example of threshold definitions on the Parameters tab of an advisor:
Time-based Thresholds

The majority of the time-based thresholds use simple duration values, such as seconds, minutes and so on. These are used to monitor such values as system uptime and, if the value for uptime drops below a certain value, indicating a restart, trigger an event.

Others use an Exponential Moving Average Window, which monitors values over a predefined time period. One such advisor is the CPU Utilization Advisor. The moving average window is used because CPU utilization can spike many times a minute, for a variety of different reasons. Raising an event for each spike would not be useful. The moving average enables you to monitor CPUs for long durations and take an average CPU utilization across that duration. Thresholds are defined against that average.

Percentage-based Thresholds

Percentage-based thresholds trigger events based on percentages of a server-defined value. Maximum number of connections, for example, raises events based on a percentage value of the total number of connections to the monitored instance or group.

Text-based Thresholds

Text-based thresholds are used to check specific configuration values are properly defined, or to retrieve success or failure messages for system processes such as backups.

19.4 Advisor Schedules

Schedules define when the Advisors collect data. There are two types of schedule:
• **Fixed Rate** — collects data according to a fixed schedule. If the schedule is set to 1 minute, and the first data collection is performed at 12:00, the subsequent data collection occurs at 12:01, even if the previous data collection is not yet complete. This is the default schedule for all Advisors.

• **Fixed Delay** — collects data only after the preceding collection is complete. If the schedule is set to 1 minute, the data collection is performed 1 minute after the preceding collection completed.

• **Disabled**: deactivates the advisor for all monitored assets, or for the selected group or host.
Chapter 20 Expression-Based Advisor Reference

Table of Contents

20.1 Administration Advisors ............................................................................................................. 187
20.2 Agent Advisors .......................................................................................................................... 193
20.3 Availability Advisors .................................................................................................................. 193
20.4 Cluster Advisors ....................................................................................................................... 195
20.5 Memory Usage Advisors .......................................................................................................... 196
20.6 Monitoring and Support Services Advisors ............................................................................... 198
20.7 Operating System Advisors ..................................................................................................... 199
20.8 Performance Advisors ............................................................................................................. 199
20.9 Replication Advisors ................................................................................................................ 204
20.10 Schema Advisors .................................................................................................................... 209
20.11 Security Advisors .................................................................................................................... 213

This chapter describes the MySQL Enterprise Monitor expression-based Advisors.

20.1 Administration Advisors

This section describes the expression-based Administration Advisors.

- 32-Bit Binary Running on 64-Bit AMD Or Intel System
- Binary Log Debug Information Disabled
- Binary Logging Is Limited
- Binary Logging Not Enabled
- Binary Logging Not Synchronized To Disk At Each Write
- Binary Logs Automatically Removed Too Quickly
- Database May Not Be Portable Due To Identifier Case Sensitivity
- Event Scheduler Disabled
- General Query Log Enabled
- Host Cache Size Not Sufficient
- In-Memory Temporary Table Size Limited By Maximum Heap Table Size
- InnoDB Status Truncation Detected
- InnoDB Strict Mode Is Off
- InnoDB Tablespace Cannot Automatically Expand
- InnoDB Transaction Logs Not Sized Correctly
- Multiple Threads Used When Repairing MyISAM Tables
- MySQL Server No Longer Eligible For Oracle Premier Support
- Next-Key Locking Disabled For InnoDB But Binary Logging Enabled
- No Value Set For MyISAM Recover Options
32-Bit Binary Running on 64-Bit AMD Or Intel System

- **Table Cache Set Too Low For Startup**
- **Time Zone Data Not Loaded**
- **Warnings Not Being Logged**

### 32-Bit Binary Running on 64-Bit AMD Or Intel System

Raises an event if a 32-bit binary is detected running on a 64-bit platform. Most 32-bit binaries can run on a 64-bit platform. However, for performance reasons, it is recommended to run 64-bit binaries on 64-bit platforms, and 32-bit binaries on 32-bit platforms.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

### Binary Logging Not Synchronized To Disk At Each Write

By default, the binary log contents are not synchronized to disk. If the server host machine or operating system crash, there is a chance that the latest events in the binary log are not persisted on disk. You can alter this behavior using the `sync_binlog` server variable. If the value of this variable is greater than 0, the MySQL server synchronizes its binary log to disk (using `fdatasync()`) after `sync_binlog` commit groups are written to the binary log. The default value of `sync_binlog` is 0, which does no synchronizing to disk - in this case, the server relies on the operating system to flush the binary log's contents from time to time as for any other file. A value of 1 is the safest choice because in the event of a crash you lose at most one commit group from the binary log. However, it is also the slowest choice (unless the disk has a battery-backed cache, which makes synchronization very fast).

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Binary Log Debug Information Disabled

The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

Binary log informational events are used for debugging and related purposes. Informational events are enabled by setting the system variable `binlog_rows_query_log_events=TRUE` (or ON). By default, this advisor generates an event if ROW or MIXED logging is enabled and `binlog_rows_query_log_events=FALSE` (or OFF).

**Note**

Binary log informational events were introduced in MySQL 5.6.2 and are not supported by earlier versions of MySQL.

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Binary Logging Is Limited

The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

Binary logging can be limited to specific databases with the `--binlog-do-db` and the `--binlog-ignore-db` options. However, if these options are used, your point-in-time recovery options are limited accordingly, along with your ability to review alterations made to your system.
Binary Logging Not Enabled

The binary log captures DML, DDL, and security changes and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

Default frequency 06:00:00
Default auto-close enabled yes

Binary Logs Automatically Removed Too Quickly

The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It is used on master replication servers as a record of the statements to be sent to slave servers. It also enables you to review all alterations made to your database.

However, the number of log files and the space they use can grow rapidly, especially on a busy server, so it is important to remove these files on a regular basis when they are no longer needed, as long as appropriate backups have been made. The expire_logs_days parameter enables automatic binary log removal.

Default frequency 12:00:00
Default auto-close enabled yes

Database May Not Be Portable Due To Identifier Case Sensitivity

The case sensitivity of the underlying operating system determines the case sensitivity of database and table names. If you are using MySQL on only one platform, you don’t normally have to worry about this. However, depending on how you have configured your server you may encounter difficulties if you want to transfer tables between platforms that differ in filesystem case sensitivity.

Default frequency 06:00:00
Default auto-close enabled yes

Event Scheduler Disabled

The Event Scheduler is a very useful feature when enabled. It is a framework for executing SQL commands at specific times or at regular intervals. Conceptually, it is similar to the idea of the Unix crontab (also known as a "cron job") or the Windows Task Scheduler.

The basics of its architecture are simple. An event is a stored routine with a starting date and time, and a recurring tag. Once defined and activated, it runs when requested. Unlike triggers, events are not linked to specific table operations, but to dates and times. Using the event scheduler, the database administrator can perform recurring events with minimal hassle. Common uses are the cleanup of obsolete data, the creation of summary tables for statistics, and monitoring of server performance and usage.

Default frequency 00:05:00
Default auto-close enabled yes

General Query Log Enabled
The general query log is a general record of what mysqld is doing. The server writes information to this log when clients connect or disconnect, and it logs each SQL statement received from clients. The general query log can be very useful when you suspect an error in a client and want to know exactly what the client sent to mysqld.

However, the general query log should not be enabled in production environments because:

- It adds overhead to the server;
- It logs statements in the order they were received, not the order they were executed, so it is not reliable for backup/recovery;
- It grows fast and can use a lot of disk space;

**Default frequency** 06:00:00  
**Default auto-close enabled** yes

### Host Cache Size Not Sufficient

The MySQL server maintains a host cache in memory that contains IP address, host name, and error information about clients. It uses the host cache for several purposes:

- By caching the results of IP-to-host name lookups, the server avoids doing a DNS lookup for each client connection, thereby improving performance.
- The cache contains information about errors that occur during the connection process. Some errors are considered "blocking." If too many of these occur successively from a given host without a successful connection, the server blocks further connections from that host.

If the host cache is not large enough to handle all the hosts from which clients may connect, performance may suffer and you may lose information about client connection errors.

**Default frequency** 00:05:00  
**Default auto-close enabled** no

### In-Memory Temporary Table Size Limited By Maximum Heap Table Size

If the space required to build a temporary table exceeds either `tmp_table_size` or `max_heap_table_size`, MySQL creates a disk-based table in the server's tmpdir directory. For performance reasons it is recommended to have most temporary tables created in memory, and only create exceedingly large temporary tables on disk.

**Default frequency** 06:00:00  
**Default auto-close enabled** yes

### InnoDB Transaction Logs Not Sized Correctly

To avoid frequent checkpoint activity and reduce overall physical I/O, which can slow down write-heavy systems, the InnoDB transaction logs should be approximately 50-100% of the size of the InnoDB buffer pool, depending on the size of the buffer pool.

**Default frequency** 06:00:00  
**Default auto-close enabled** yes

### InnoDB Status Truncation Detected
InnoDB primarily uses the SHOW ENGINE INNODB STATUS command to dump diagnostics information. As this SHOW statement can output a lot of data when running in a system with very many concurrent sessions, the output is limited to 64 kilobytes in versions < 5.5.7, and 1 megabyte on versions greater than 5.5.7. You are running a version where the truncation limit should be 1 megabyte, however truncation is still occurring in your system, and the MEM Agent relies on this output to pass back a number of key InnoDB statistics.

However, InnoDB provides a startup option called innodb_status_file, which dumps the same output as SHOW ENGINE INNODB STATUS to a file called innodb_status.<mysql pid> in the datadir. The MEM Agent (in versions > 2.3.0) reads this file automatically, if it exists, before executing the SHOW statement.

Default frequency 00:05:00
Default auto-close enabled no

InnoDB Strict Mode Is Off

To guard against ignored typos and syntax errors in SQL, or other unintended consequences of various combinations of operational modes and SQL commands, InnoDB provides a "strict mode" of operations. In this mode, InnoDB raises error conditions in certain cases, rather than issue a warning and process the specified command (perhaps with some unintended defaults). This is analogous to MySQL’s sql_mode, which controls what SQL syntax MySQL accepts, and determines whether it silently ignores errors, or validates input syntax and data values.

Using the new clauses and settings for ROW_FORMAT and KEY_BLOCK_SIZE on CREATE TABLE and ALTER TABLE commands and the CREATE INDEX command can be confusing when not running in strict mode. Unless you run in strict mode, InnoDB ignores certain syntax errors and creates the table or index, with only a warning in the message log. However if InnoDB strict mode is on, such errors generate an immediate error and the table or index is not created, thus saving time by catching the error at the time the command is issued.

Default frequency 12:00:00
Default auto-close enabled yes

InnoDB Tablespace Cannot Automatically Expand

If the InnoDB tablespace is not allowed to automatically grow to meet incoming data demands and your application generates more data than there is room for, out-of-space errors occur and your application may experience problems.

Default frequency 06:00:00
Default auto-close enabled yes

Multiple Threads Used When Repairing MyISAM Tables

Using multiple threads when repairing MyISAM tables can improve performance, but it can also lead to table and index corruption.

Default frequency 06:00:00
Default auto-close enabled yes

MySQL Server No Longer Eligible For Oracle Premier Support

To ensure you are running versions of MySQL which are still covered by their support contracts, this advisor checks for MySQL versions which are no longer eligible for Premier support cover. Specifically, for versions 5.1 and 5.5.
The default thresholds are defined in a numeric format, where version 5.5 is represented as 50500 (Notice threshold), and 5.1 as 50100 (warning threshold).

**Default frequency** 06:00:00

**Default auto-close enabled** yes

### Next-Key Locking Disabled For InnoDB But Binary Logging Enabled

Next-key locking in InnoDB can be disabled, which may improve performance in some situations. However, this may result in inconsistent data when recovering from the binary logs in replication or recovery situations. You can disable most gap locks, including most next-key locks, by using `--transaction-isolation=READ-COMMITTED` or `--innodb_locks_unsafe_for_binlog=1`. Using either is perfectly safe, but only if you are also using `--binlog-format=ROW`.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

### No Value Set For MyISAM Recover Options

The `myisam-recover-options` option (named `myisam-recover` before MySQL 5.5.3) enables automatic MyISAM crash recovery should a MyISAM table become corrupt for some reason. If this option is not set, tables are “Marked as crashed” if they are corrupt, and no sessions are able to SELECT or perform any sort of DML against it.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

### Table Cache Set Too Low For Startup

The table cache size controls the number of open tables that can occur at any one time on the server. MySQL opens and close tables as needed, however you should avoid having the table cache set too low, causing MySQL to constantly open and close tables to satisfy object access.

If the table cache limit has been exceeded by the number of tables opened in the first three hours of service, then the table cache size is likely set too low.

**Default frequency** 00:30:00

**Default auto-close enabled** yes

### Time Zone Data Not Loaded

The MySQL server supports multiple time zones and provides various date and time functions, including a function that converts a datetime value from one time zone to another (`CONVERT_TZ`). However, while the MySQL installation procedure creates the time zone tables in the mysql database, it does not load them; you must do so manually after installation. If the time zone tables are not loaded, certain time zone functions such as `CONVERT_TZ` will not work.

**Default frequency** 12:00:00

**Default auto-close enabled** yes

### Warnings Not Being Logged

Error conditions encountered by a MySQL server are always logged in the error log, but warning conditions are only logged if `log_warnings` is set to a value greater than 0. If warnings are not
logged, valuable information about aborted connections and various other communication errors is not stored. This is especially important if you use replication so you get more information about what is happening, such as messages about network failures and reconnection.

Default frequency 12:00:00
Default auto-close enabled yes

20.2 Agent Advisors

This section describes the expression-based Agent Advisors.

- MySQL Agent Memory Usage Excessive
- MySQL Agent Not Reachable

MySQL Agent Memory Usage Excessive

The memory needed by the MySQL Agent for basic monitoring is fairly small and consistent, and depends on the number of rules you have enabled. However, when the Query Analyzer is enabled, the Agent can use significantly more memory to monitor and analyze whatever queries you direct through it. In this case, the amount of memory used depends on the number of unique normalized queries, example queries and example explains being processed, plus the network bandwidth required to send query data to the Service Manager. In general, the amount of memory used for the Query Analyzer is small and well-bounded, but under some circumstances it can become excessive, especially on older versions of Linux.

Default frequency 00:01:00
Default auto-close enabled no

MySQL Agent Not Reachable

In order to monitor a MySQL server, a Service Agent must be running and communicating with the Service Manager. If the Agent cannot communicate with the Service Manager, the Service Manager has no way of knowing if the MySQL database server being monitored is running, and it cannot collect current statistics to properly evaluate the rules scheduled against that server.

Default frequency 00:00:01
Default auto-close enabled yes

20.3 Availability Advisors

This section describes the expression-based Availability Advisors.

- Attempted Connections To The Server Have Failed
- Excessive Percentage Of Attempted Connections To The Server Have Failed
- Maximum Connection Limit Nearing Or Reached
- MySQL Availability
- MySQL Server Has Been Restarted

MySQL Availability

Tracks MySQL availability, by making a full connection to the monitored instance on the configured frequency.
Maximum Connection Limit Nearing Or Reached

Once the maximum connection limit for the MySQL server has been reached, no other user connections can be established and errors occur on the client side of the application.

Default frequency 00:05:00

Default auto-close enabled yes

MySQL Server Has Been Restarted

To perform useful work, a database server must be up-and-running continuously. It is normal for a production server to run continuously for weeks, months, or longer. If a server has been restarted recently, it may be the result of planned maintenance, but it may also be due to an unplanned event that should be investigated.

Default frequency 00:05:00

Default auto-close enabled no

Excessive Percentage Of Attempted Connections To The Server Have Failed

Excess aborted connection attempts to MySQL may indicate an issue with respect to the server or network, or could be indicative of DoS or password-cracking attempts against the MySQL Server. The aborted-connects count is incremented when:

• A client does not have privileges to access a database
• A client uses the wrong password
• A malformed packet is received
• The connect_timeout variable is exceeded

Default frequency 00:05:00

Default auto-close enabled no

Attempted Connections To The Server Have Failed

Aborted connection attempts to MySQL may indicate an issue with respect to the server or network, or could be indicative of DoS or password-cracking attempts against the MySQL Server. The aborted-connects count is incremented when:

• A client does not have privileges to access a database
• A client uses the wrong password
• A malformed packet is received
• The connect_timeout variable is exceeded
Cluster Data Node Data Memory Getting Low
Advises when the amount of Data Memory configured for the data nodes starts to run low. Database inserts start to fail as all of the memory is consumed.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Has Been Restarted
To perform useful work, the cluster data nodes must be up-and-running continuously. It is normal for a production system to run continuously for weeks, months, or longer. If a data node has been restarted recently, it may be the result of planned maintenance, but it may also be due to an unplanned event that should be investigated.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Index Memory Getting Low
Advises when the amount of Index Memory configured for the data nodes starts to run low. Database inserts fail as all of the memory is consumed.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Redo Buffer Space Getting Low
Advises when the redo buffers start to fill up.

Default frequency 00:05:00
Cluster Data Node Redo Log Space Getting Low

Advises when the redo log spaces start to fill up.
Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Undo Buffer Space Getting Low

Advises when the undo buffers start to fill up.
Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Undo Log Space Getting Low

Advises when the undo log spaces start to fill up.
Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Nodes Not Running

Indicates how many data nodes are not running.
Default frequency 00:05:00
Default auto-close enabled no

Cluster DiskPageBuffer Hit Ratio Is Low

Advises when the hit-rate for the DiskPageBuffer falls below a threshold. May happen temporarily after restarting one or more data nodes. This is the average ratio since the last sample period.
Default frequency 00:05:00
Default auto-close enabled no

Cluster Has Stopped

Indicates a cluster has completely stopped.
Default frequency 00:02:00
Default auto-close enabled no

20.5 Memory Usage Advisors

This section describes the expression-based Memory Usage Advisors.

- InnoDB Buffer Cache Has Sub-Optimal Hit Rate
- Key Buffer Size May Not Be Optimal For Key Cache
- Query Cache Has Sub-Optimal Hit Rate
InnoDB Buffer Cache Has Sub-Optimal Hit Rate

Logical I/O is many times faster than physical I/O, and therefore a DBA should strive to keep physical I/O to a minimum. It is true that logical I/O is not free, and that the DBA should work to keep all I/O to a minimum, but it is best if most data access is performed in memory. When using InnoDB, most data access should occur in RAM, and therefore the InnoDB buffer cache hit rate should be high.

Default frequency 00:05:00
Default auto-close enabled no

Key Buffer Size May Not Be Optimal For Key Cache

The key cache hit ratio represents the proportion of keys that are being read from the key cache in memory instead of from disk. This should normally be greater than 99% for optimum efficiency.

Default frequency 00:05:00
Default auto-close enabled no

Query Cache Has Sub-Optimal Hit Rate

When enabled, the query cache should experience a high degree of "hits", meaning that queries in the cache are being reused by other user connections. A low hit rate may mean that not enough memory is allocated to the cache, identical queries are not being issued repeatedly to the server, or that the statements in the query cache are invalidated too frequently by INSERT, UPDATE or DELETE statements.

This advisor triggers when more than 25% of the Query Cache is being used, and the ratio of Query Cache hits to Query Cache inserts is low.

Default frequency 00:05:00
Default auto-close enabled no

Query Cache Potentially Undersized

When the Query Cache is full, and needs to add more queries to the cache, it makes more room in the cache by freeing the least recently used queries from the cache, and then inserting the new queries. If this is happening often then you should increase the size of the cache to avoid this constant "swapping".

Default frequency 00:05:00
Default auto-close enabled no

Table Cache Not Optimal

MySQL is multi-threaded, so there may be many clients issuing queries for a given table simultaneously. To minimize the problem with multiple client threads having different states on the same table, the table is opened independently by each concurrent thread.

The table cache is used to cache file descriptors for open tables and there is a single cache shared by all clients. Increasing the size of the table cache enables mysqld to keep more tables open.
simultaneously by reducing the number of file open and close operations that must be done. If the value of `open_tables` is approaching the value of `table_cache`, this may indicate performance problems.

**Default frequency** 00:05:00

**Default auto-close enabled** no

### Thread Cache Size May Not Be Optimal

Each connection to the MySQL database server runs in its own thread. Thread creation takes time, so rather than killing the thread when a connection is closed, the server can keep the thread in its thread cache and use it for a new connection later.

**Default frequency** 00:05:00

**Default auto-close enabled** no

### 20.6 Monitoring and Support Services Advisors

This section describes the Monitoring and Support Services Advisors.

- **HTTP Server Performance**
- **Service Manager Health**
- **Support Diagnostics**
- **Wrong Version Agent Tracker**

### Wrong Version Agent Tracker

Tracks wrong version agents that try to connect to this service manager.

### HTTP Server Performance

Provides instruments for data that exposes the performance of an HTTP server.

### Service Manager Health

Provides instruments for data that exposes the performance of MySQL Enterprise Service Manager.

This advisor is responsible for the following:

- Provides the data for the graphs on the MEM Service Manager page. To display these graphs, select the **MEM Service Manager** item in the Asset Selector for your MySQL Enterprise Service Manager in the All Timeseries Graphs page.

- Checks the timestamps of data collected by the agent to ensure the time of the monitored server is not set to a future time or date. Any data collected, with a timestamp of more than 5 minutes in the future, relative to the MySQL Enterprise Service Manager's system clock, is discarded and a critical event is generated. The critical event contains information on the assets whose time is incorrectly defined.

**Important**

It is strongly recommended you ensure your MySQL Enterprise Service Manager server and all monitored instances synchronize their system clocks with the same time server.
• Raises a critical event if the SMTP Rate Limit defined on an Event Handler is exceeded. If this rate is exceeded, no further notifications are sent until the period ends and the new period begins (1 minute). The event lists the name of the event handler whose rate limit was exceeded and the rate defined on that event handler.

These events are not auto-closed and are not updated. That is, they only display the first failure.

To create an event handler which sends notifications when the SMTP Rate Limit is exceeded, in the Create Event Handler window, select the ServiceManager: MEM Service Manager asset and the Critical Event Status. Define other values as required.

![Important]

This can result in a very large volume of emails, depending on the SMTP Rate Limits defined on your Event Handlers.

Support Diagnostics

Tracks MySQL configuration for bundling in the support diagnostics.

20.7 Operating System Advisors

These advisors are described in Chapter 21, GUI-Based Advisor Reference.

20.8 Performance Advisors

This section describes the Performance Advisors.

• Binary Log Usage Exceeding Disk Cache Memory Limits
• Excessive Disk Temporary Table Usage Detected
• Excessive Number of Locked Processes
• Excessive Number of Long Running Processes
• Excessive Number of Long Running Processes Locked
• Flush Time Set To Non-Zero Value
• Indexes Not Being Used Efficiently
• InnoDB Buffer Pool Writes May Be Performance Bottleneck
• InnoDB Flush Method May Not Be Optimal
• InnoDB Log Buffer Flushed To Disk After Each Transaction
• InnoDB Not Using Newest File Format
• InnoDB Log Waits May Be Performance Bottleneck
• MyISAM Concurrent Insert Setting May Not Be Optimal
• Prepared Statements Not Being Closed
• Prepared Statements Not Being Used Effectively
• Query Cache Is Excessively Fragmented
• Table Lock Contention Excessive
Binary Log Usage Exceeding Disk Cache Memory Limits

When binary log usage exceeds the binary log cache memory limits, it is performing excessive disk operations. For optimal performance, transactions that move through the binary log should be contained within the binary log cache.

**Default frequency** 00:05:00

**Default auto-close enabled** no

Excessive Disk Temporary Table Usage Detected

If the space required to build a temporary table exceeds either `tmp_table_size` or `max_heap_table_size`, MySQL creates a disk-based table in the server's tmpdir directory. Also, tables that have TEXT or BLOB columns are automatically placed on disk.

For performance reasons it is ideal to have most temporary tables created in memory, leaving exceedingly large temporary tables to be created on disk.

**Default frequency** 00:05:00

**Default auto-close enabled** no

Excessive Number of Locked Processes

Depending on the circumstances, storage engines, and other factors, one process may be using or accessing a resource (e.g. a table or row) required by another process in such a way that the second process cannot proceed until the first process releases the resource. In this case the second process is in a "locked" state until the resource is released. If many processes are in a locked state it may be a sign of serious trouble related to resource contention, or a long running session that is not releasing currently held locks when it should have.

**Default frequency** 00:01:00

**Default auto-close enabled** no

Excessive Number of Long Running Processes

Most applications and databases are designed to execute queries very quickly. If many queries are taking a long time to execute (e.g. more than a few seconds) it can be a sign of trouble. In such cases queries may need to be tuned or rewritten, or indexes added to improve performance. In other cases the database schema may have to be redesigned.

**Default frequency** 00:01:00

**Default auto-close enabled** no

Excessive Number of Long Running Processes Locked

Most applications and databases are designed to execute queries very quickly, and to avoid resource contention where one query is waiting for another to release a lock on some shared resource. If many queries are locked and taking a long time to execute (e.g. more than a few seconds), it can be a sign
of performance trouble and resource contention. In such cases queries may need to be tuned or rewritten, or indexes added to improve performance. In other cases the database schema may have to be redesigned.

**Default frequency** 00:01:00

**Default auto-close enabled** no

### Flush Time Set To Non-Zero Value

If `flush_time` is set to a non-zero value, all tables are closed every `flush_time` seconds to free up resources and synchronize unflushed data to disk. If your system is unreliable and tends to lock up or restart often, forcing out table changes this way degrades performance but can reduce the chance of table corruption or data loss. We recommend that this option be used only on Windows, or on systems with minimal resources.

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Indexes Not Being Used Efficiently

The target server does not appear to be using indexes efficiently. The values of `Handler_read_rnd_next` and `Handler_read_rnd` together - which reflect the number of rows read via full table scans - are high compared to the `Handler` variables which denote index accesses - such as `Handler_read_key`, `Handler_read_next` etc. You should examine your tables and queries for proper use of indexes.

**Default frequency** 00:05:00

**Default auto-close enabled** no

### InnoDB Buffer Pool Writes May Be Performance Bottleneck

For optimal performance, InnoDB should not have to wait before writing pages into the InnoDB buffer pool.

**Default frequency** 00:05:00

**Default auto-close enabled** yes

### InnoDB Flush Method May Not Be Optimal

Different values for `innodb_flush_method` can have a marked effect on InnoDB performance. In some versions of GNU/Linux and Unix, flushing files to disk by invoking `fsync()` (which InnoDB uses by default) or other similar methods, can be surprisingly slow. If you are dissatisfied with database write performance, you might try setting the `innodb_flush_method` parameter to `O_DIRECT` or `O_DSYNC`.

**Default frequency** 06:00:00

**Default auto-close enabled** no

### InnoDB Log Buffer Flushed To Disk After Each Transaction

By default, InnoDB's log buffer is written out to the log file at each transaction commit and a flush-to-disk operation is performed on the log file, which enforces ACID compliance. In the event of a crash, if you can afford to lose a second's worth of transactions, you can achieve better performance by setting `innodb_flush_log_at_trx_commit` to either 0 or 2. If you set the value to 2, then only an operating system crash or a power outage can erase the last second of transactions. This can be very
useful on slave servers, where the loss of a second's worth of data can be recovered from the master server if needed.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

### InnoDB Not Using Newest File Format

InnoDB supports compressed tables (**COMPRESSED** row format) and more efficient BLOB handling (**DYNAMIC** row format), but both features require support for the latest file format (**innodb_file_format=Barracuda**). These features also require the use of the **ROW_FORMAT=[DYNAMIC|COMPRESSED]** in **CREATE TABLE** and **ALTER TABLE** statements.

**Default frequency** 12:00:00

**Default auto-close enabled** no

### InnoDB Log Waits May Be Performance Bottleneck

For optimal performance, InnoDB should not have to wait before writing DML activity to the InnoDB log buffer.

**Default frequency** 00:05:00

**Default auto-close enabled** no

### MyISAM Concurrent Insert Setting May Not Be Optimal

MyISAM uses table-level locking, which can adversely affect performance when there are many concurrent INSERT and SELECT statements because INSERTs blocks all SELECTs until the INSERT is completed. However, MyISAM can be configured to allow INSERT and SELECT statements to run concurrently in certain situations.

- If **concurrent_insert is set to 1** (the default, or **AUTO** as of MySQL 5.5.3 or later), MySQL allows INSERT and SELECT statements to run concurrently for **MyISAM tables that have no free blocks in the middle of the data file**.

- If **concurrent_insert is set to 2** (available in MySQL 5.0.6 and later, or **ALWAYS** as of MySQL 5.5.3 or later), MySQL allows concurrent inserts for all **MyISAM tables**, even those that have holes. For a table with a hole, new rows are inserted at the end of the table if it is in use by another thread. Otherwise, MySQL acquires a normal write lock and inserts the row into the hole.

Setting concurrent_insert to 2 allows tables to grow even when there are holes in the middle. This can be bad for applications that delete large chunks of data but continue to issue many SELECTs, thus effectively preventing INSERTs from filling the holes.

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Prepared Statements Not Being Closed

Prepared statements may increase performance in applications that execute similar statements more than once, primarily because the query is parsed only once. Prepared statements can also reduce network traffic because it is only necessary to send the data for the parameters for each execution rather than the whole statement.

However, prepared statements take time to prepare and consume memory in the MySQL server until they are closed, so it is important to use them properly. If you are not closing prepared statements
when you are done with them, you are needlessly tying up memory that could be put to use in other ways.

Default frequency 00:05:00

Default auto-close enabled no

Prepared Statements Not Being Used Effectively

Prepared statements may increase performance in applications that execute similar statements more than once, primarily because the query is parsed only once. Prepared statements can also reduce network traffic because it is only necessary to send the data for the parameters for each execution rather than the whole statement.

However, prepared statements take time to prepare and consume memory in the MySQL server until they are closed, so it is important to use them properly. If you are only executing a statement a few times, the overhead of creating a prepared statement may not be worthwhile.

Default frequency 00:05:00

Default auto-close enabled no

Query Cache Is Excessively Fragmented

Enabling the query cache can significantly increase performance for SELECT queries that are identically executed across many connections, returning the same result set. However, performance can be adversely affected if the memory used for the query cache is excessively fragmented, causing the server to pause while it is removing entries from the cache or searching the free block list for a good block to use to insert a new query into the cache.

Default frequency 00:05:00

Default auto-close enabled no

Table Lock Contention Excessive

Performance can be degraded if the percentage of table operations that have to wait for a lock is high compared to the overall number of locks. This can happen when using a table-level locking storage engine, such as MyISAM, instead of a row-level locking storage engine.

Default frequency 00:05:00

Default auto-close enabled no

Thread Cache Not Enabled

Each connection to the MySQL database server runs in its own thread. Thread creation takes time, so rather than killing the thread when a connection is closed, the server can keep the thread in its thread cache and use it for a new connection later.

Default frequency 00:05:00

Default auto-close enabled no

Thread Pool Stall Limit Too Low

The thread_pool_stall_limit variable enables the thread pool to handle long-running statements. If a long-running statement was permitted to block a thread group, all other connections assigned to the group would be blocked and unable to start execution until the long-running statement completed. In the worst case, this could take hours or even days.
The value of `thread_pool_stall_limit` should be chosen such that statements that execute longer than its value are considered stalled. Stalled statements generate a lot of extra overhead since they involve extra context switches and in some cases even extra thread creations. On the other hand, setting the `thread_pool_stall_limit` parameter too high means long-running statements block short-running statements for longer than necessary. Short wait values permit threads to start more quickly. Short values are also better for avoiding deadlock situations. Long wait values are useful for workloads that include long-running statements, to avoid starting too many new statements while the current ones execute.

**Default frequency 00:05:00**

**Default auto-close enabled no**

**Thread Pooling Not Enabled**

As of MySQL 5.5.16, commercial distributions of MySQL include a thread pool plugin that provides an alternative thread-handling model designed to reduce overhead and improve performance. It implements a thread pool that increases server performance by efficiently managing statement execution threads for large numbers of client connections.

With servers that have many concurrent active connections (generally, more than the number of CPUs within the machine) it can be beneficial for performance to enable the Thread Pool plugin. This keeps the number of actively executing threads within the server lower, generally leaving less contention for locks and resources, whilst still maintaining very high connection counts from applications.

**Default frequency 00:05:00**

**Default auto-close enabled no**

**Too Many Concurrent Queries Running**

Too many active queries indicates there is a severe load on the server, and may be a sign of lock contention or unoptimized SQL queries.

**Default frequency 00:05:00**

**Default auto-close enabled no**

**20.9 Replication Advisors**

This section describes the Replication Advisors.

- Binary Log Checksums Disabled
- Binary Log File Count Exceeds Specified Limit
- Binary Log Row Based Images Excessive
- Binary Log Space Exceeds Specified Limit
- Replication Configuration Advisor
- Replication Status Advisor
- Master Not Verifying Checksums When Reading From Binary Log
- Slave Detection Of Network Outages Too High
- Slave Execution Position Too Far Behind Read Position
- Slave Has Login Accounts With Inappropriate Privileges
Binary Log Checksums Disabled

Binary logs written and read by the MySQL Server are now crash-safe, because only complete events (or transactions) are logged or read back. By default, the server logs the length of the event as well as the event itself and uses this information to verify that the event was written correctly.

You can also cause the server to write checksums for the events using CRC32 checksums by setting the `binlog_checksum` system variable, to add an extra level of safety to the logs and the replication process. To cause the server to read checksums from the binary log, use the `master_verify_checksum` system variable. The `slave_sql_verify_checksum` system variable causes the slave SQL thread to read checksums from the relay log.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

Binary Log File Count Exceeds Specified Limit

The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables replication as well as point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database. However, binary logs consume disk space and file system resources, and can be removed from a production server after they are no longer needed by the slaves connecting to this master server, and after they have been backed up.

**Default frequency** 06:00:00

**Default auto-close enabled** no

Binary Log Row Based Images Excessive

As of MySQL Server 5.6, row-based replication now supports row image control. By logging only those columns required for uniquely identifying and executing changes on each row (as opposed to all columns) for each row change, it is possible to save disk space, network resources, and memory usage. You can determine whether full or minimal rows are logged by setting the `binlog_row_image` server system variable to one of the values `minimal` (log required columns only), `full` (log all columns), or `noblob` (log all columns except for unneeded BLOB or TEXT columns).

**Default frequency** 06:00:00

**Default auto-close enabled** yes

Binary Log Space Exceeds Specified Limit
The binary log is a set of files that contain information about data modifications made by the MySQL server. It enables replication as well as point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

However, binary logs can consume a very large amount of disk space and should be removed from a production server to free up space after they are no longer needed by the slaves connecting to this master server, and after they have been backed up.

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Replication Configuration Advisor

Analyzes the configuration of masters and slaves in replication topologies and alerts when configuration problems have been detected:

- More than one server has the same value for server_id (duplicate server IDs)
- The max_allowed_packet size on a slave is less than its master
- When a master is replicating to a slave that has an older version of the MySQL Server than the master

### Replication Status Advisor

Monitors slave replication status and alerts when replication has stopped or is compromised in some way (e.g. one of the slave threads has stopped), displays the last error messages seen, and where possible provides specific advice to fix the errors.

### Master Not Verifying Checksums When Reading From Binary Log

Binary logs written and read by the MySQL Server are now crash-safe, because only complete events (or transactions) are logged or read back. By default, the server logs the length of the event as well as the event itself and uses this information to verify that the event was written correctly.

You can also cause the server to write checksums for the events using CRC32 checksums by setting the `binlog_checksum` system variable, to add an extra level of safety to the logs and the replication process. To cause the server to read checksums from the binary log, use the `master_verify_checksum` system variable. The `slave_sql_verify_checksum` system variable causes the slave SQL thread to read checksums from the relay log.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

### Slave Detection Of Network Outages Too High

Slaves must deal with network connectivity outages that affect the ability of the slave to get the latest data from the master, and hence cause replication to fall behind. However, the slave notices the network outage only after receiving no data from the master for `slave_net_timeout` seconds. You may want to decrease `slave_net_timeout` so the outages -- and associated connection retries -- are detected and resolved faster. The default for this parameter is 3600 seconds (1 hour), which is too high for many environments.

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Slave Execution Position Too Far Behind Read Position
When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave’s SQL thread reads the relay logs and executes the updates they contain. If the position from which the SQL thread is reading is way behind the position to which the I/O thread is currently writing, it is a sign that replication is getting behind and results of queries directed to the slave may not reflect the latest changes made on the master.

Default frequency 00:05:00
Default auto-close enabled no

Slave Has Login Accounts With Inappropriate Privileges

Altering and dropping tables on a slave can break replication. Unless the slave also hosts non-replicated tables, there is no need for accounts with these privileges. As an alternative, you should set the read_only flag ON so the server allows no updates except from users that have the SUPER privilege or from updates performed by slave threads.

Default frequency 06:00:00
Default auto-close enabled no

Slave Master Info/Relay Log Info Not Crash Safe

MySQL now supports logging of master connection information and of slave relay log information to tables as well as files. In order for replication to be crash-safe, that information must be logged to tables and those tables must each use a transactional storage engine such as InnoDB.

Default frequency 06:00:00
Default auto-close enabled yes

Slave Not Configured As Read Only

Arbitrary or unintended updates to a slave may break replication or cause a slave to be inconsistent with respect to its master. Making a slave read_only can be useful to ensure that a slave accepts updates only from its master server and not from clients; it minimizes the possibility of unintended updates.

Default frequency 06:00:00
Default auto-close enabled no

Slave Not Verifying Checksums When Reading From Relay Log

Binary logs written and read by the MySQL Server are now crash-safe, because only complete events (or transactions) are logged or read back. By default, the server logs the length of the event as well as the event itself and uses this information to verify that the event was written correctly.

You can also cause the server to write checksums for the events using CRC32 checksums by setting the binlog_checksum system variable, to add an extra level of safety to the logs and the replication process. To cause the server to read checksums from the binary log, use the master_verify_checksum system variable. The slave_sql_verify_checksum system variable causes the slave SQL thread to read checksums from the relay log.

Default frequency 06:00:00
Default auto-close enabled yes

Slave Relay Log Space Is Very Large
When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. After the SQL thread has executed all the updates in a relay log, the file is no longer needed and can be deleted to conserve disk space.

**Default frequency** 06:00:00

**Default auto-close enabled** no

---

### Slave Relay Logs Not Automatically Purged

When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. After the SQL thread has executed all the updates in a relay log, the file is no longer needed and can be deleted to conserve disk space.

**Default frequency** 06:00:00

**Default auto-close enabled** no

---

### Slave SQL Processing Not Multi-Threaded

As of MySQL Server version 5.6, replication now supports parallel execution of transactions with multi-threading on the slave. When parallel execution is enabled, the slave SQL thread acts as the coordinator for a number of slave worker threads as determined by the value of the `slave_parallel_workers` server system variable.

The current implementation of multi-threading on the slave assumes that data and updates are partitioned on a per-database basis, and that updates within a given database occur in the same relative order as they do on the master. However, it is not necessary to coordinate transactions between different databases. Transactions can then also be distributed per database, which means that a worker thread on the slave can process successive transactions on a given database without waiting for updates to other databases to complete.

Transactions on different databases can occur in a different order on the slave than on the master, simply checking for the most recently executed transaction is not a guarantee that all previous transactions on the master have been executed on the slave. This has implications for logging and recovery when using a multi-threaded slave.

Finally, note that beginning with MySQL Server 5.7.2, there is also support for intra-schema parallelization (LOGICAL_CLOCK). See Replication Slave Options and Variables for more information.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

---

### Slave SQL Thread Reading From Older Relay Log Than I/O Thread

When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. If the SQL thread is reading from an older relay log than the one to which the I/O thread is currently writing, it is a sign that replication is getting behind and results of queries directed to the slave may not reflect the latest changes made on the master.

**Default frequency** 00:05:00

**Default auto-close enabled** no

---

### Slave Too Far Behind Master
If a slave is too far behind the master, results of queries directed to the slave may not reflect the latest changes made on the master.

**Default frequency** 00:01:00  
**Default auto-close enabled** yes

### Slave Without REPLICATION SLAVE Accounts

If the master ever fails, you may want to use one of the slaves as the new master. An account with the REPLICATION SLAVE privilege must exist for a server to act as a replication master (so a slave can connect to it), so it's a good idea to create this account on your slaves to prepare it to take over for a master if needed.

**Default frequency** 06:00:00  
**Default auto-close enabled** no

### 20.10 Schema Advisors

This section describes the Schema advisors.

- AUTO_INCREMENT Field Limit Nearly Reached
- Object Changed: Database Has Been Altered
- Object Changed: Database Has Been Created
- Object Changed: Database Has Been Dropped
- Object Changed: Function Has Been Created
- Object Changed: Function Has Been Dropped
- Object Changed: Index Has Been Created
- Object Changed: Index Has Been Dropped
- MyISAM Indexes Found with No Statistics
- Object Changes Detected
- Server-Enforced Data Integrity Checking Disabled
- Server-Enforced Data Integrity Checking Not Strict
- Object Changed: Table Has Been Altered
- Object Changed: Table Has Been Created
- Object Changed: Table Has Been Dropped
- Tables Found with No Primary or Unique Keys
- Object Changed: User Has Been Dropped

### AUTO_INCREMENT Field Limit Nearly Reached

Many applications need to generate unique numbers and sequences for identification purposes (e.g. customer IDs, bug or trouble ticket tags, membership or order numbers, etc). MySQL's mechanism for doing this is the AUTO_INCREMENT column attribute, which enables you to generate sequential numbers automatically.
However, the range of numbers that can be generated is limited by the underlying data type. For example, the maximum value possible for a TINYINT UNSIGNED column is 255. If you try to generate a number that exceeds the maximum allowed by the underlying data type (e.g. by inserting a NULL value into the AUTO_INCREMENT column), database errors occur and your application may behave unexpectedly.

The primary purpose of AUTO_INCREMENT in MySQL is to generate a sequence of positive integers. The use of non-positive numbers in an AUTO_INCREMENT column is unsupported, so you may as well define those columns to be UNSIGNED, which effectively doubles their allowable range.

**Default frequency** 06:00:00

**Default auto-close enabled** no

---

**Object Changed: Database Has Been Altered**

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

**Default frequency** 00:10:00

**Default auto-close enabled** no

---

**Object Changed: Database Has Been Created**

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

**Default frequency** 00:10:00

**Default auto-close enabled** no

---

**Object Changed: Database Has Been Dropped**

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

**Default frequency** 00:10:00

**Default auto-close enabled** no

---

**Object Changed: Function Has Been Created**

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures or functions and investigate the reasons for the changes.

**Default frequency** 00:10:00

**Default auto-close enabled** no

---

**Object Changed: Function Has Been Dropped**

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures or functions and investigate the reasons for the changes.
Object Changed: Index Has Been Created

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

Default frequency 00:10:00
Default auto-close enabled no

Object Changed: Index Has Been Dropped

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

Default frequency 00:10:00
Default auto-close enabled no

MyISAM Indexes Found with No Statistics

The MySQL optimizer needs index statistics to help make choices about whether to use indexes to satisfy SQL queries. Having no statistics or outdated statistics limits the optimizer's ability to make smart and informed access plan choices.

Default frequency 12:00:00
Default auto-close enabled no

Object Changes Detected

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

Default frequency 00:10:00
Default auto-close enabled no

Server-Enforced Data Integrity Checking Disabled

SQL Modes define what SQL syntax MySQL should support and what kind of data validation checks it should perform. If no SQL modes are enabled this means there is no form of server-enforced data integrity, which means invalid incoming data is not be rejected by the server, but is instead changed to conform to the target column's default datatype.

Note
Any client can change its own session SQL mode value at any time.

Default frequency 06:00:00
Default auto-close enabled no

Server-Enforced Data Integrity Checking Not Strict
SQL Modes define what SQL syntax MySQL should support and what kind of data validation checks it should perform. There are many possible options that can be used in conjunction with each other to specify varying degrees of syntax and data validation checks the MySQL server will perform. However, to ensure the highest level of confidence for data integrity, at least one of the following should be included in the list: TRADITIONAL, STRICT_TRANS_TABLES, or STRICT_ALL_TABLES.

**Note**
Any client can change its own session SQL mode value at any time.

Default frequency 06:00:00
Default auto-close enabled no

Object Changed: Table Has Been Altered

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

Default frequency 00:10:00
Default auto-close enabled no

Object Changed: Table Has Been Created

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

Default frequency 00:10:00
Default auto-close enabled no

Object Changed: Table Has Been Dropped

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

Default frequency 00:10:00
Default auto-close enabled no

Tables Found with No Primary or Unique Keys

A primary or unique key of a relational table uniquely identifies each record in the table. Except in very unusual circumstances, every database table should have one or more columns designated as the primary key or as a unique key, and it is common practice to declare one.

**Note**
Tables lacking primary or unique keys can have a very negative impact on replication performance when using binlog-format=ROW.

Default frequency 12:00:00
Default auto-close enabled no
Object Changed: User Has Been Dropped

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when changes occur in a production environment with respect to database structures and investigate the reasons for the changes.

Default frequency 00:10:00

Default auto-close enabled no

20.11 Security Advisors

This section describes the Security Advisors.

- Account Has An Overly Broad Host Specifier
- Account Has Global Privileges
- Account Has Old Insecure Password Hash
- Account Has Strong MySQL Privileges
- Account Requires Unavailable Authentication Plug-ins
- Insecure Password Authentication Option Is Enabled
- Insecure Password Generation Option Is Enabled
- LOCAL Option Of LOAD DATA Statement Is Enabled
- Non-root User Has GRANT Privileges On All Databases
- Non-root User Has Server Admin Privileges
- Non-root User Has DB, Table, Or Index Privileges On All Databases
- Policy-Based Password Validation Does Not Perform Dictionary Checks
- Policy-Based Password Validation Is Weak
- Policy-Based Password Validation Not Enabled
- Privilege Alterations Detected: Privileges Granted
- Privilege Alterations Detected: Privileges Revoked
- Privilege Alterations Have Been Detected
- Root Account Can Login Remotely
- Root Account Without Password
- SHA-256 Password Authentication Not Enabled
- Server Contains Default "test" Database
- Server Has Accounts Without A Password
- Server Has Anonymous Accounts
- Server Has No Locally Authenticated Root User
- Server Includes A Root User Account
Account Has An Overly Broad Host Specifier

The MySQL server has user accounts with overly broad host specifiers. A MySQL account is identified by both a username and a hostname, which are found in the User and Host columns of the mysql.user table. The User value is the name that a client must supply when connecting to the server. The Host value indicates the host or hosts from which the user is allowed to connect. If this is a literal hostname, the account is limited to connections only from that host. If the hostname contains the '%' wildcard character, the user can connect from any host that matches the wildcard character and potentially from any host at all.

From a security standpoint, literal host values are best and % is worst. Accounts that have Host values containing wildcards are more susceptible to attack than accounts with literal host values, because attackers can attempt to connect from a broader range of machines.

For example, if an account has user and host values of root and %, it means that you can connect as the root user from any machine if you know the password. By contrast, if the host name is localhost or 127.0.0.1, the attacker can only attempt to connect as the root user from the server host.

Default frequency 00:05:00
Default auto-close enabled no

Account Has Global Privileges

A MySQL server may have user accounts with privileges on all databases and tables (*, *). In most cases global privileges should be allowed only for the MySQL root user, and possibly for users that you trust or use for backup purposes. Global privileges such as DROP, ALTER, DELETE, UPDATE, INSERT, and LOCK TABLES may be dangerous as they may cause other users to be affected adversely.

Default frequency 00:05:00
Default auto-close enabled no

Account Has Strong MySQL Privileges

Certain account privileges can be dangerous and should only be granted to trusted users when necessary. For example, the FILE privilege allows a user to read and write files on the database server (which includes sensitive operating system files), the PROCESS privilege allows currently executing statements to be monitored, and the SHUTDOWN privilege allows a user to shut down the server. In addition, the GRANT privilege allows a user to grant privileges to others.

Default frequency 00:05:00
Default auto-close enabled no

Account Requires Unavailable Authentication Plug-ins

MySQL supports many forms of authentication as of the 5.5 release, including external authentication mechanisms using PAM, or Windows native authentication with commercial releases of MySQL version
5.5.16 or greater. If a user is configured to use an authentication plugin, and that plugin is not loaded on server start, access to the database is blocked for those users.

**Default frequency 06:00:00**

**Default auto-close enabled yes**

### Account Has Old Insecure Password Hash

Prior to MySQL 4.1, password hashes computed by the `PASSWORD()` function were 16 bytes long. As of MySQL 4.1 (and later), `PASSWORD()` was modified to produce a longer 41-byte hash value to provide enhanced security.

**Default frequency 06:00:00**

**Default auto-close enabled no**

### Insecure Password Authentication Option Is Enabled

Prior to MySQL 4.1, password hashes computed by the `PASSWORD()` function were 16 bytes long. As of MySQL 4.1 (and later), `PASSWORD()` was modified to produce a longer 41-byte hash value to provide enhanced security. However, in order to allow backward-compatibility with user tables that have been migrated from pre-4.1 systems, you can configure MySQL to accept logins for accounts that have password hashes created using the old, less-secure `PASSWORD()` function, but this is not recommended.

**Default frequency 06:00:00**

**Default auto-close enabled no**

### Insecure Password Generation Option Is Enabled

Prior to MySQL 4.1, password hashes computed by the `PASSWORD()` function were 16 bytes long. As of MySQL 4.1 (and later), `PASSWORD()` was modified to produce a longer 41-byte hash value to provide enhanced security. In order to allow backward-compatibility with older client programs, you can configure MySQL to generate short (pre-4.1) password hashes for new passwords, however, this is not recommended.

**Default frequency 06:00:00**

**Default auto-close enabled no**

### LOCAL Option Of LOAD DATA Statement Is Enabled

The `LOAD DATA` statement can load a file that is located on the server host, or it can load a file that is located on the client host when the `LOCAL` keyword is specified.

There are two potential security issues with supporting the `LOCAL` version of `LOAD DATA` statements:

- The transfer of the file from the client host to the server host is initiated by the MySQL server. In theory, a patched server could be built that would tell the client program to transfer a file of the server’s choosing rather than the file named by the client in the `LOAD DATA` statement. Such a server could access any file on the client host to which the client user has read access.

- In a Web environment where the clients are connecting from a separate web server, a user could use `LOAD DATA LOCAL` to read any files that the web server process has read access to (assuming that a user could run any statement against the SQL server). In this environment, the client with respect to the MySQL server actually is the web server, not the remote program being run by the user who connects to the web server.
Non-root User Has GRANT Privileges On All Databases

The GRANT privilege, when given on all databases as opposed to being limited to a few specific databases, enables a user to give to other users those privileges that the grantor possesses on all databases. It can be used for databases, tables, and stored routines. Such a privilege should be limited to as few users as possible. Users who do indeed need the GRANT privilege should have that privilege limited to only those databases they are responsible for, and not for all databases.

Default frequency 00:05:00
Default auto-close enabled no

Non-root User Has Server Admin Privileges

Certain privileges, such as SHUTDOWN and SUPER, are primarily used for server administration. Some of these privileges can have a dramatic effect on a system because they allow someone to shutdown the server or kill running processes. Such operations should be limited to a small set of users.

Default frequency 01:00:00
Default auto-close enabled no

Non-root User Has DB, Table, Or Index Privileges On All Databases

Privileges such as SELECT, INSERT, ALTER, and so forth allow a user to view and change data, as well as impact system performance. Such operations should be limited to only those databases to which a user truly needs such access so the user cannot inadvertently affect other people's applications and data stores.

Default frequency 01:00:00
Default auto-close enabled no

Policy-Based Password Validation Is Weak

When users create weak passwords (e.g. 'password' or 'abcd') it compromises the security of the server, making it easier for unauthorized people to guess the password and gain access to the server. Starting with MySQL Server 5.6, MySQL offers the 'validate_password' plugin that can be used to test passwords and improve security. With this plugin you can implement and enforce a policy for password strength (e.g. passwords must be at least 8 characters long, have both lowercase and uppercase letters, and contain at least one special non-alphanumeric character).

Default frequency 06:00:00
Default auto-close enabled no

Policy-Based Password Validation Does Not Perform Dictionary Checks

When users create weak passwords (e.g. 'password' or 'abcd') it compromises the security of the server, making it easier for unauthorized people to guess the password and gain access to the server. Starting with MySQL Server 5.6, MySQL offers the 'validate_password' plugin that can be used to test passwords and improve security. With this plugin you can implement and enforce a policy for password strength (e.g. passwords must be at least 8 characters long, have both lowercase and uppercase letters, contain at least one special non-alphanumeric character, and do not match commonly-used words).
Policy-Based Password Validation Not Enabled

When users create weak passwords (e.g. 'password' or 'abcd') it compromises the security of the server, making it easier for unauthorized people to guess the password and gain access to the server. Starting with MySQL Server 5.6, MySQL offers the 'validate_password' plugin that can be used to test passwords and improve security. With this plugin you can implement and enforce a policy for password strength (e.g. passwords must be at least 8 characters long, have both lowercase and uppercase letters, and contain at least one special non-alphanumeric character).

Privilege Alterations Have Been Detected

For development environments, changes to database security privileges may be a normal occurrence, but for production environments it is wise to know when any security changes occur with respect to database privileges, and to ensure that those changes are authorized and required.

Root Account Can Login Remotely

By default, MySQL includes a root account with unlimited privileges that is typically used to administer the MySQL server. If possible, accounts with this much power should not allow remote logins in order to limit access to only those users able to login to the machine on which MySQL is running. This helps prevent unauthorized users from accessing and changing the system.
The root user account has unlimited privileges and is intended for administrative tasks. Privileged accounts should have strong passwords to prevent unauthorized users from accessing and changing the system.

**Default frequency** 00:05:00

**Default auto-close enabled** yes

### Server Has Accounts Without A Password

Accounts without passwords are particularly dangerous because an attacker needs to guess only a username. Assigning passwords to all accounts helps prevent unauthorized users from accessing the system.

**Default frequency** 00:05:00

**Default auto-close enabled** yes

### Server Has Anonymous Accounts

Anonymous MySQL accounts allow clients to connect to the server without specifying a username. Since anonymous accounts are well known in MySQL, removing them helps prevent unauthorized users from accessing the system.

**Default frequency** 00:05:00

**Default auto-close enabled** yes

### Server Has No Locally Authenticated Root User

MySQL 5.5 supports both built-in authentication and external authentication via other methods such as PAM (LDAP, Unix user authentication) and Windows native authentication. However, if all 'root' users are configured to use external authentication, and this external authentication were to fail (such as the LDAP server losing power), all administrator access to the MySQL Server is denied.

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Server Includes A Root User Account

By default, MySQL includes a root account with unlimited privileges that is typically used to administer the MySQL server. There is no reason this account must be named 'root'. Accounts with this much power should not be easily discovered. Since the root account is well known in MySQL, changing its name helps prevent unauthorized users from accessing and changing the system.

**Default frequency** 00:05:00

**Default auto-close enabled** no

### Server Contains Default "test" Database

By default, MySQL comes with a database named test that anyone can access. This database is intended only for testing and should be removed before moving into a production environment. Because the default test database can be accessed by any user and has permissive privileges, it should be dropped immediately as part of the installation process.

**Default frequency** 00:05:00

**Default auto-close enabled** no
SHA-256 Password Authentication Not Enabled

To help keep the server secure, each user's password is encrypted, and the stronger the encryption method, the more secure the server is. Starting with MySQL Server 5.6, MySQL offers a new encryption algorithm that performs authentication using SHA-256 password hashing. This is stronger encryption than that available with native authentication (i.e. the standard encryption method).

**Default frequency** 06:00:00

**Default auto-close enabled** no

Symlinks Are Enabled

You can move tables and databases from the database directory to other locations and replace them with symbolic links to the new locations. You might want to do this, for example, to move a database to a file system with more free space or to increase the speed of your system by spreading your tables to different disks.

However, symlinks can compromise security. This is especially important if you run mysqld as root, because anyone who has write access to the server's data directory could then delete any file in the system!

**Default frequency** 06:00:00

**Default auto-close enabled** no

User Has Rights To Database That Does Not Exist

When a database is dropped, user privileges on the database are not automatically dropped. This has security implications as that user regains privileges if a database with the same name is created in the future, which may not be the intended result.

**Default frequency** 00:05:00

**Default auto-close enabled** no

User Has Rights To Table That Does Not Exist

When a table is dropped, user privileges on the table are not automatically dropped. This has security implications as that user regains privileges if a table with the same name in the same database is created in the future, which may not be the intended result.

**Default frequency** 00:05:00

**Default auto-close enabled** no

Users Can View All Databases On MySQL Server

The SHOW DATABASES privilege should be granted only to users who need to see all the databases on a MySQL Server. It is recommended that the MySQL Server be started with the `--skip-show-database` option enabled to prevent anyone from using the SHOW DATABASES statement unless they have been specifically granted the SHOW DATABASES privilege.

**Note**

If a user is granted any global privilege, such as CREATE TEMPORARY TABLES or LOCK TABLES, they are automatically given the ability to show databases unless the server is started with the `--skip-show-database` option enabled. DBAs should be aware of this fact, in the event that any applications make use of temporary tables.
Default frequency 00:05:00
Default auto-close enabled no
Chapter 21 GUI-Based Advisor Reference

Table of Contents

21.1 Agent Health Advisor 221
21.2 MySQL Enterprise Backup Health Advisor 222
21.3 MySQL Process Discovery Advisor 224
21.4 Duplicate MySQL Server UUID 225
21.5 CPU Utilization Advisor 225
21.6 Filesystem Free Space Advisor 227
21.7 Query Analysis Advisors 228

This chapter describes the MySQL Enterprise Monitor GUI-based Advisors. That is, the advisors which are configured using a dialog rather than an expression.

21.1 Agent Health Advisor

The Agent Health Advisor monitors the monitoring agent's resource usage, communication status, backlog and memory usage.

The Agent Health Advisor configuration dialog is divided into the following functional areas:

- General
- Communication
- Backlog

General

The General section defines the CPU and RAM usage thresholds. These thresholds generate events if the defined threshold value is broken by either CPU or RAM usage. Both threshold definitions use a moving average window. Although it is possible to use very small values for a moving average window, large values, larger than seconds, are recommended.

Figure 21.1 Agent Health - General
- **Agent CPU Threshold**: enables you to define thresholds for percentage CPU usage. The default value is Critical at 10% usage.

- **Memory Usage Thresholds (% of max allowed)**: enables you to define thresholds for RAM usage as a percentage of the maximum heap size allocated to the monitoring agent. The default values are:
  - Notice = 70
  - Warning = 85
  - Critical = 90

**Communication**

The Communication section defines the thresholds for latency and HTTP errors between agent and MySQL Enterprise Service Manager.

**Figure 21.2 Agent Health - General**

- **Agent Latency Thresholds**: enables you to define thresholds for time difference between the time the data was collected and the time the MySQL Enterprise Service Manager received the collected data. This can be caused by clocks that are not synchronized, network problems, and so on. The default values are:
  - Warning = 1 minute
  - Critical = 10 minutes

**Important**

Under certain circumstances, such as MySQL Enterprise Service Manager experiencing heavy load, events can be raised for **Agent host time out of sync relative to dashboard**. These can occur even though both MySQL Enterprise Service Manager and the monitored host are synchronized with the same time server and no time-synchronization problems exist.

The Agent Health Advisor compares the time on the MySQL Enterprise Service Manager against the time on the monitored host. If no time-synchronization issues exist, these false positive events are auto-closed.
• **HTTP Error Thresholds (% of total requests):** enables you to define thresholds for number of HTTP errors as a percentage of the total number of HTTP requests. The default values are:
  
  • Notice = 10
  • Warning = 20
  • Critical = 30

**Backlog**

If the monitoring agent is unable to communicate with the MySQL Enterprise Service Manager, it stores the collected data in memory up to a limit of 10MB, then on the filesystem, up to a limit of 10MB, giving a total limit of backlog storage of 20MB. If the limit is reached, backlog data is dropped.

**Figure 21.3 Agent Health - Backlog**

• **Backlog Memory Usage Thresholds (% of max allowed):** enables you to define a threshold for the amount of RAM used by the backlog, as a percentage of the maximum RAM allowed, 10MB. The default value is Warning = 80, which corresponds to 8MB of RAM used.

• **Backlog Disk Usage Thresholds (% of max allowed):** enables you to define a threshold for the amount of disk space used by the backlog, as a percentage of the maximum disk space allowed, 10MB. The default value is Warning = 80, which corresponds to 8MB of disk space used.

**21.2 MySQL Enterprise Backup Health Advisor**

This section describes the MySQL Enterprise Backup Health Advisor which checks the status of backups, and alerts according to whether they succeeded or failed.

• **Notify on succeeded or failed backups:** enables you to generate an event for the success or failure of a backup. The default values are:
  
  • Notice = Success
  • Emergency = Failure

There are no other return types.

• **Notify when incremental backups are not being used:** enables you to generate an event if the monitoring agent detects that incremental backups are not used. Select **Yes** to generate an event.

• **Notify when backup lock time is excessive:** enables you to generate an event if the backup lock time exceeds the defined thresholds. The default values are:
MySQL Process Discovery Advisor

- Notice = 10 seconds
- Warning = 1 minute

- **Notify when the age of the last backup is too old:** enables you to generate an event if the last backup is older than the defined threshold. The default value is:
  - Warning = 7 days

An event is generated for each backup run. Each event is identified by the MySQL Enterprise Backup backup ID. Successful backup events are auto-closed.

If a backup fails, generating a failure event, but a subsequent backup using the same command line is successful, both the success and failure event are auto-closed.

### 21.3 MySQL Process Discovery Advisor

The **MySQL Process Discovery** Advisor enables you to find and, optionally, establish a connection with unmonitored MySQL instances. If you choose not to attempt a connection with the discovered instances, they are listed in the **Unmonitored MySQL Instances** list on the **MySQL Instances** dashboard.

![Important](image)

If you disable this advisor, notifications for unmonitored instances, and the associated events, are not displayed in the user interface.

#### Table 21.1 MySQL Process Discovery Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attempt Connection</td>
<td>Whether or not to attempt a connection. If this is set to No, the advisor continues to raise events related to unmonitored instances. If set to Yes, a connection is attempted using the credentials supplied.</td>
</tr>
<tr>
<td>Alert Level</td>
<td>Level of alert generated if an unmonitored instance is discovered.</td>
</tr>
<tr>
<td>Admin User</td>
<td>The root user of the instance, or a user that has the SUPER, CREATE and INSERT privileges on the schema in which the inventory table is created. The inventory table stores unique identifiers for the instance, and is created in the <code>mysql</code> schema by default. The SUPER privilege is required to temporarily switch off replication when creating and populating the inventory table. If you choose to enable the <strong>Auto-Create Less Privileged Users</strong> option, this user is used to create those with the required privileges to monitor this instance. In this case, it also requires the PROCESS, REPLICATION CLIENT, SELECT and SHOW DATABASES privileges globally WITH GRANT OPTION.</td>
</tr>
<tr>
<td>Admin Password</td>
<td>The password for the Admin User.</td>
</tr>
</tbody>
</table>
| Auto-Create Less Privileged Users | When monitoring an instance, multiple levels of user can be employed to ensure that a Process connection is not held open indefinitely.  
  - **General User:** used for general monitoring tasks that do not require SUPER level privileges, and is always connected.  
  - **Limited User:** used for potentially long running statements running with SELECT only privileges. |
### Duplicate MySQL Server UUID

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General User</td>
<td>This user handles general monitoring tasks that do not require SUPER level privileges. Lower privileged users are used, unless higher privileges are required. In which case we temporarily log in as the SUPER privileged user, and then fall back to the general user. If you are manually managing this user, it should have at least the PROCESS, REPLICATION CLIENT, SELECT and SHOW DATABASES privileges globally.</td>
</tr>
<tr>
<td>General Password</td>
<td>The password for the user with general privileges.</td>
</tr>
<tr>
<td>Limited User</td>
<td>This user is used for statements that are limited to a single connection, and can be run with global SELECT privileges. Examples of these kinds of statements include retrieving database metadata from INFORMATION_SCHEMA tables, or any custom SQL that is used to monitor application specific statistics. If you are manually managing this user, it should have at least the SELECT and SHOW DATABASES privileges globally.</td>
</tr>
<tr>
<td>Limited Password</td>
<td>The password for the user with limited privileges.</td>
</tr>
<tr>
<td>MySQL Instance Identity Source</td>
<td>Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Default</strong>: uses either the <code>server_uuid</code> variable, if present, or generates a random new identity.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Host plus Data Directory</strong>: uses a hash of the host identity and the path to the MySQL instances data directory to create a unique identity. If you are manually managing this user, it should have at least the SELECT and SHOW DATABASES privileges globally.</td>
</tr>
</tbody>
</table>

Default auto-close enabled yes

### 21.4 Duplicate MySQL Server UUID

Tracks instances whose UUID is duplicated or becomes associated with multiple, different host names, or connections, over a specific time period. These changes are measured by rate, that is, by a defined number of changes over the defined time period.

- **Change Rate**: number of changes per time frame.
- **Every**: time frame in which the changes are tracked.

For example, if the **Change Rate** is set to 5, and **Every** set to 10 minutes, and the UUID of the instance changed hostname 5 times in 8 minutes, an event is generated.

Default auto-close enabled yes

### 21.5 CPU Utilization Advisor

Monitors and graphs CPU usage on the monitored server or groups of servers.

**CPU Usage**
**CPU Outliers**

Enables detection of CPU outliers. A CPU is considered an outlier if the conditions defined here are met.

**Moving Average Window**: duration of the Exponential Moving Average (EMA) window. For more information on EMA, see [Time-based Thresholds](#).

**CPU Usage Thresholds**: configure the thresholds for percentage of total CPU usage.

**CPI IO Wait Thresholds**: configure the thresholds for CPU IO Wait as a percentage of total CPU time.

Default auto-close enabled yes

**CPU Outliers**

Enables detection of CPU outliers. A CPU is considered an outlier if the conditions defined here are met.

**Note**

It is not recommended to enable this for all Operating Systems, but for specific groups.

**Enable CPU Outlier Detection**: Whether or not to enable the CPU outlier detection.

**Minimum Server Count for Outlier Detection**: Minimum required sample size before outlier detection is enabled.
• **Small Group Notification**: Whether or not to generate an event if the sample size is too small to enable outlier detection.

• **Outlier Percentile**: percentage, relative to the other CPUs in the group, at which a CPU is considered an outlier.

## 21.6 Filesystem Free Space Advisor

Monitors and graphs the filesystem disk space usage.

**Default auto-close enabled** yes

### General

![General section](image)

The **General** section enables you to choose the filesystem to monitor. The following types are available:

- **Local Disk**: enables monitoring of the local hard disks.
- **Network Mount**: enables monitoring of mounted network filesystems on the monitored server.
- **RAM Disk**: enables monitoring of RAM disks configured on the server.
- **CDROM**: enables monitoring of CD-ROM drives on the server
- **Swap**: enables monitoring of the system's swap file.

Select the filesystem types, as required, from the drop-down list.

To remove a filesystem type, click the x on the filesystem label.

### Estimated Full Capacity

![Estimated Full Capacity section](image)

The **Estimated Full Capacity** section monitors and graphs the time remaining to full capacity based on existing load.
• **Extrapolate Free Space to Zero in Graphs**: enables graphing of the projected time to full capacity, based on existing load.

• **Free Space Running Out Thresholds**: generate events based on when the free space is projected to run out.

**Percentage of Space**

**Figure 21.8 Filesystem - Percentage of Space**

The **Percentage of Space** section generates events based on the percentage of free space available, relative to the total space on the monitored device.

**Percentage Used in Time Range**

**Figure 21.9 Filesystem - Percentage Used in Time Range**

Monitors the percentage of disk space consumed per unit of time.

**21.7 Query Analysis Advisors**

This section describes the **Query Analysis** advisors.

**Average Statement Execution Time Advisor**

Monitors the average execution time of a normalized SQL statement and generates events if the execution time exceeds the defined thresholds.

This advisor has the following parameters:

• **Average Execution Time Thresholds**: Generates events if the average execution time exceeds the defined thresholds.

• **Minimum Execution Count**: Minimum number of times a normalized statement must be executed before it can generate an event.

• **One Alert per Query**: Specify how events are generated. The possible values are:

  • **Yes**: generate an event for each normalized query that exceeds a threshold
Query Pileup Advisor

- **No**: generate a single event per MySQL Server summarizing all queries that exceed the thresholds. This is the default behavior.

- **DML Statements Only**: Specify for which statements events are generated. The possible values are:
  - **Yes**: generate events for DML statements only.
  - **No**: generate events for all SQL statements.

**Query Pileup Advisor**

Alerts when query pileups occur, when the number of threads running increase rapidly over a short period of time. For example, based on the defaults for this advisor, if the exponential moving average of Threads_running has increased by 50% or more, but less than 80%, over the last 1 minute, it raises a Warning alert.

- **Window Size**: duration of the moving average window over which monitoring is done.

- **Growth Rate Thresholds**: percentage growth rate of the running statements during the defined moving average window.

- **Minimum Running Threads**: the minimum number of running threads before an event is generated.

**SQL Statement Generates Warnings or Errors**

Generates events when a normalized SQL statement generates errors or warnings over a period of time.

- **One Alert Per Query**: generate events for queries which return errors or warnings. Possible values are:
  - **Yes**: generate an event for each normalized query which returns an error or warning.
  - **No**: generate a single event, per MySQL server, summarizing all queries which generated errors or warnings.

**Query Analysis Reporting**

Enables capturing and reporting of query analysis data.

- **Enable Example Query**: provides detailed data about the queries and their parameters. Enabling this parameter results in an increase in the RAM used by the monitoring agent.

Important

This feature requires `events_statement_history_long` be enabled in `performance_schema.setup_consumers`. This is disabled by default in MySQL 5.6.

- **Enable Example Explain**: executes EXPLAIN on the selected statement. This is executed for statements whose runtime exceeds the value defined in **Auto-Explain Threshold**.

- **Auto-Explain Threshold**: Explains are executed for statements whose runtime is longer than the value defined here.

Important

Explains are generated for query data supplied by the MySQL Enterprise Monitor Proxy and Aggregator, Connector/J plugin, and Performance Schema sources.
Explain is supported for all DML statements on MySQL 5.6.3 or higher. On earlier versions, only `SELECT` is supported.
You can customize your MySQL Enterprise Monitor rules, advisors, and graphs, based on your organization's business rules, best practices, and the types of issues you can anticipate.

22.1 Customizing Groups

Groups are central to how MySQL Enterprise Monitor operates. For example, an Asset (such as a host or MySQL instance) added to a group automatically inherits all Advisors scheduled for that group.

**Note**

In MySQL Enterprise Monitor 2.3, schedules were set per Asset, and not per group.

Each panel in the Service Manager enables you to distinguish between groups (e.g., graph a specific group), and groups help configure the handling of Events and Advisors.

MySQL Enterprise Monitor has two different types of server groups:

- **Ad hoc**: You manually construct these groups, which are typically grouped using hosts and MySQL instances. For example, you might define "Development" and "Production" server groups.

- **Replication**: MEM automatically generates a grouping for the replication topology, if present.

**Note**

The text is captured as-is in the text field when defining a group name. For example, HTML entities are not converted.

The following example shows two Server groups, Development and Production:
22.2 Creating Advisors and Rules

For common scenarios, reuse or edit the advisors and graphs provided by MySQL Enterprise. To create new advisors and graphs for your own needs, go to the Configuration on top menu bar and choose the Advisors menu item, select the Create Advisor button on the General Advisors Control or select the Import/Export button to create a graph.

22.2.1 Creating Advisors

Similar existing Advisors are grouped together in Advisor category. To create a new Advisor, go to Configuration on top menu bar and choose the Advisors menu item, select the Create Advisor button on the General Advisors Control.

Default Advisor Categories

The following are the default Advisor categories:

- Administration
- Agent
- Availability
- Backup
- Cluster
- Graphing
- Memory Usage
- Monitoring and Support Services
- Operating System
- Performance
- Query Analysis
- Replication
- Schema

For more information about configuring groups, see Section 14.4, “MySQL Instances Dashboard”.

---

**Figure 22.1 Manage Instances example**

![MySQL Enterprise Monitor](image-url)

For more information about configuring groups, see Section 14.4, “MySQL Instances Dashboard”.

---

Copyright © 2005, 2013, Oracle and/or its affiliates. All rights reserved. 320-0-2880 - scissors (112.168.1.215) - Aug 20, 2013 9:07:04 pm (1b Since: 1 day, 3 hours ago) - About

---

232
• Security

Note
You can also create your own Advisor category while creating an Advisor by changing the Advisor Category to a custom value.

22.2.2 Overview of Graph Creation

Graphs are defined using XML, and then imported into MEM. The new custom graph is displayed with the default graphs, sorted by name on the graphs page.

For an example of how to create a graph, see Section 22.2.9, “Creating a New Graph: An Example”.

The XML elements for creating a graph are as follows:

• version
  The version number of the graph. Generally only important with the bundled graphs, and is only used internally.

• uuid
  The unique id of the graph. Each revision (version) requires a new uuid, which is only used internally.

• name
  The visible graph name, which is displayed within the graph listing. Note: graphs are sorted alphabetically.

• frequency
  Optionally define the frequency for the graph, which defaults to 1 minute. May use seconds, minutes, hours, and days.

• rangeLabel
  The Y-axis range label. For example, a graph about disk space usage may use MB.

• series
  Each series contains a label and an expression. The label is the visible name of the series, and the simple expression defines it.

• variables
  Each variables definition contains a name, instance, and dcItem element. The instance defines what data the graph displays, and each dcItem element contains a nameSpace, className, and attribName:

  • nameSpace
    Namespace (type) of the data collection item.

  • className
    Class (namespace type) of the data collection item.

  • attribName
    Attribute name of the data collection item.

See the listing of data collection items for available nameSpace, className, and attribName items.
22.2.3 Overview of Advisor Creation

To create a new Advisor with all-new settings, click the **Create Advisor** button available on the Advisors page. To create an Advisor similar to an existing one, click the Advisor menu drop-down icon to the left of the Advisor title, and choose the **Copy Advisor** menu item. You can edit any Advisor element during the copying process, unlike editing an existing Advisor. You can also delete an existing Advisor created by you, click the Advisor menu drop-down icon to the left of the Advisor title, and choose click the **Delete Advisor** menu item.

You can change the Advisor name, change the Advisor category that an Advisor belongs to, set your own version number, and alter the threshold and frequency of an Advisor.

---

**Note**

If you do not specify a version number for the new Advisor, the version 1.0 is automatically added. Most importantly, you can alter an Advisor’s expression. Expressions are the core of a MySQL Enterprise Advisor and define the scenario being monitored. An expression can be as simple as a single server parameter or can be complex, combining multiple parameters with mathematical operations.

Most importantly, you can alter an Advisor’s expression. Expressions are the core of a MySQL Enterprise Advisors and define the scenario being monitored. An expression can be as simple as a single server parameter or can be complex, combining multiple parameters with mathematical operations.

An expression has two main characteristics:

- An expression tests whether a best practice is being violated.
- The result of an expression must always be 1 or 0 (corresponding to true or false).

For example, if you decide that enabling binary logging is a best practice for a production server (as Oracle recommends), then this best practice is violated if `log_bin` is OFF. Consequently, the expression for the “Binary Logging Not Enabled” rule is “%log_bin% == OFF”. If this evaluates to 1, an event 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 product uses the Java Expression Parser. The operators and functions consist of:

- The **IN()** operator.
- The MySQL functions **LEAST(), LOCATE(), ABS(), MOD(), NOW()** (returns time since Unix epoch UTC in seconds), **UNIX_TIMESTAMP** (technically a no-op), and **INTERVAL [n] SECOND, MINUTE, HOUR, WEEK, MONTH**.
- Comparisons with MySQL timestamps and datetimes collected by the agent in the standard MySQL format 'YYYY-MM-DD hh:mm:ss[.nanos]'.
- The **IF** function: `IF (condition, true_expression, false_expression)` returns either `true_expression` or `false_expression`, depending on whether `condition` is true or false. This function uses short-circuit evaluation, so only one of the return expressions is evaluated.
- The **LEFT(string, length)** and **RIGHT(string, length)** functions.
- The **NUM(string)** function.
Variables

Note
The `CAST(expression as type)` function is not implemented. Instead, use `NUM(string)` to use strings as numbers.

- Other functions and operators may be implemented as needed to assist with custom rule creation. Open a service request if you have such a requirement.

For a complete list of the built-in variables used to create Advisors, see Server Option and Variable Reference.

Creating an expression is dependent on variables defined in the Variable Assignment frame. This frame links variables used in the expression field with data gathered from the target MySQL server instance: server status variables, operating system status information, and table information. Variable names are associated with elements in the Data Item drop-down menu. To define more than one variable, click the add row button. For a complete listing of the data collection items used in creating rules, see Appendix G, Data Collection Items.

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

Note
When saving a new Advisor, choose a unique name not used by any existing Advisor.

22.2.4 Variables

When MySQL Enterprise Monitor evaluates an expression, it replaces variables with values. For example, part of the expression for the “MyISAM Key Cache Has Sub-Optimal Hit Rate” rule calculates the hit rate as follows:

\[ 100 - \left( \frac{\%Key\_reads\%}{\%Key\_read\_requests\%} \right) \times 100 \]

If the current value of \%Key\_reads\% is 4522 and the current value of \%Key\_read\_requests\% is 125989, the hit ratio is 96.4%:

\[ 100 - \left( \frac{4522}{125989} \right) \times 100 \]

By convention, the Advisors supplied by MySQL use ‘%’ as the delimiter, for example, \%Key\_reads\%. This makes variables more readily identifiable.

Variables can be used in the Description, Advice, Action, and Links attributes of a rule, as well as in expressions. This lets you report the current value of an expression. For instance, you can add the message, “The current value of Key\_reads is \%Key\_reads\%.” to the Advice text box. When this is displayed on the screen, the value of \%Key\_reads\% is substituted into the text. If \%Key\_reads\% has a value of 4522, the message becomes “The current value of Key\_reads is 4522.”

22.2.5 Thresholds

Each expression has a threshold value that triggers an alert. The `THRESHOLD` keyword associates that value with an alert level: either an Notice, Warning, or Critical alert.

For example, the expression for the performance advisor, “Thread Cache Size May Not Be Optimal”, is:

\[ 100 - \left( \frac{\%Threads\_created\%}{\%Connections\%} \right) \times 100 < \text{THRESHOLD} \]
The **Threshold** is set at 95% for an Info level alert, 85% for a Warning alert, and 75% for a Critical alert, producing alerts of three different levels.

Expressions can be straightforward. The expression for "Binary Logging Not Enabled" (one of the Administration alerts) is:

```
%log_bin% == Threshold
```

When the result is **OFF**, only one alert is triggered: a Warning level alert. You cannot just use the expression `%log_bin% == "OFF"`, because this would not test binary logging against a threshold and so would not result in an alert.

Specify precise conditions when each expression should be evaluated, to avoid false alarms. For example, the expression for the “MyISAM Key Cache Has Sub-Optimal Hit Rate” rule is:

```
(%Uptime% > 10800) && (%Key_read_requests% > 10000) && (100-((%Key_reads% / %Key_read_requests%) * 100) < Threshold)
```

The first part of the expression, `(%Uptime% > 10800)`, delays evaluating this expression until the system has been running for 10800 seconds (3 hours). When a server starts up, it might take a while to reach a state that is representative of normal operations. For example, the InnoDB buffer pool, MyISAM key cache, and the SQL query cache might require some time to fill up with application data, after which the cached data boosts performance.

In addition, if some part of the system is not heavily used, an alert might 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 could be triggered based on very limited use of other MyISAM tables such as the `mysql.user` table. For this reason, this advisor has a second part: `(%Key_read_requests% > 10000)`. The rule is not evaluated unless there is plenty of activity associated with the key cache.

### 22.2.6 Using Strings

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

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

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

When the expression is evaluated, either "OFF" or "ON" is substituted for `%Slave_running%`, and "Yes" or "No" for `%Slave_IO_Running%`, depending on the state of your system. If the slave is running but the I/O thread is not, the expression becomes:

```
("ON" == "ON") && ("No" != "Yes")
```

Without quotation marks, this expression would not evaluate to **TRUE** as it should.

**Note**

So that it is interpreted properly, the `==` operator is converted to `=` before being passed to the MySQL expression parser.

### 22.2.7 Wiki Format

When editing or defining a rule, you can enter text in Wiki format in the Problem Description, Advice, Recommended Action, and Links and Further Reading text boxes. You can format and highlight text and add hyperlinks, using the notation listed in the following table.
### Table 22.1 MySQL Enterprise Monitor: Wiki Formatting

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>boldface text</td>
</tr>
<tr>
<td><del>italic</del></td>
<td>italicize text</td>
</tr>
<tr>
<td>\</td>
<td>create a line break</td>
</tr>
<tr>
<td>\</td>
<td>create a double line break</td>
</tr>
</tbody>
</table>
| \\
|                | create a backslash           |
| *item 1*        | create a bulleted list item  |
| #item 1          | create a numbered list item  |
| \               | use the `\` to escape special characters |
| '{moreInfo:name|url}' | create a hyperlink          |

So the following Wiki text:

Replication is a **very nice feature** of MySQL. Replication can be very useful for solving problems in the following areas:
* Data Distribution
* Load Balancing
* Backup and Recovery
You can check replication status and start a slave using the following commands: SHOW SLAVE STATUS \G\START SLAVE; '{moreInfo:MySQL Manual: Replication FAQ|http://dev.mysql.com/doc/refman/5.6/en/faqs-replication.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:
* Data Distribution
* Load Balancing
* Backup and Recovery
You can check replication status and start a slave with the following commands: SHOW SLAVE STATUS \G;START SLAVE; <a href="http://dev.mysql.com/doc/refman/5.6/en/faqs-replication.html" target="_blank">MySQL Manual: Replication FAQ</a>

### 22.2.8 Creating a New Advisor: An Example

This section documents the steps to create an Advisor.

To create an Advisor, select the Create Advisor button from the Advisors page. The new advisor page is displayed:
This example creates an Advisor that checks if connections have been killed using the KILL statement and generates an event.

Create your custom rule by following these steps:

1. Using the Advisor Name text box, give the Advisor an appropriate name, such as "Connections killed".
2. From the Advisor Category drop down list box, choose an Advisor category for your Advisor.
3. Define the variable for your expression in the Variable Assignment frame.
   - In the Variable text box, enter %connections_killed%, the variable used in the Expression text box.
   - In the Data Item drop-down list, select the mysql:status:Com_kill entry. For a description of all the data items available, see Appendix G, Data Collection Items.
   - In the Instance text box, enter local.
4. Enter the following expression in the Expression text area.

   '%connections_killed% > THRESHOLD'

5. Set the following threshold:
Creating a New Graph: An Example

- Set the **Info Alert** level to 0. An informational event is generated if 1 or more connections are killed.

6. Add appropriate entries for the **Problem Description**, **Advice**, and **Links** text areas. Optionally, use Wiki markup for these text areas. You can also reference the `%connections_killed%` variable in these text areas.

7. Save the Advisor

After you create the Advisor, schedule it against the MySQL server you want to monitor. For instructions on Configure Advisor, see Table 19.3, “Advisor Edit Menu Controls”.

### 22.2.9 Creating a New Graph: An Example

This section documents the steps to create a graph. Before creating a graph, review the preceding sections of this chapter as Graphs and Rules use similar components. And for an overview that’s specific to graphs, see Section 22.2.2, “Overview of Graph Creation”

This example creates a graph that checks and compares disk usage, by displaying the usage and total available disk space over time.

Begin by navigating to the **Configuration, Advisors** page, and click the **Import/Export** link. Then note the **Custom Rule/Graph/Data Items Import** section. This is where the XML file is imported.

A definition to check disk space usage may look like the following:

```xml
<?xml version="1.0"?>
<com_mysql_merlin_server_graph_Design>
  <version>1.0</version>
  <uuid>a57c2bba-ea9b-102b-b396-94aca32bee29</uuid>
  <name>my filename usage test</name>
  <rangeLabel>MB</rangeLabel>
  <series>
    <label>used</label>
    <expression>used_fs/1024/1024</expression>
  </series>
  <series>
    <label>total size</label>
    <expression>total_fs/1024/1024</expression>
  </series>
  <variables>
    <name>used_fs</name>
    <dcItem>
      <nameSpace>os</nameSpace>
      <className>fs</className>
      <attribName>fs_used</attribName>
    </dcItem>
    <instance>/</instance>
  </variables>
  <variables>
    <name>total_fs</name>
    <dcItem>
      <nameSpace>os</nameSpace>
      <className>fs</className>
      <attribName>fs_total</attribName>
    </dcItem>
    <instance>/</instance>
  </variables>
</com_mysql_merlin_server_graph_Design>
```

Upon successfully loading a graph, a popup notification may say "1 graph imported" in the MySQL Enterprise Monitor User Interface.
Custom Data Collection

This also creates a new Advisor with the same title, which is unscheduled by default. Go to Configuration, Advisors, Graphing to locate and enable this new Advisor.

This graph is displayed on the appropriate graphs page (like every other graph) under the name defined within the definition, which is "my filename usage test" in the example above.

22.3 Custom Data Collection

This section describes how to configure custom data collections for the monitoring agent.

The monitoring agent can be configured to collect data directly from the MySQL server, using a query. This enables you to extend the functionality of the agent and create custom advisors which analyze the data collected by the custom data collection.

To create a custom data collection, you must add a class to custom.xml, located in the etc directory of your agent installation. Each defined class is a custom data collection.

**Note**
custom.xml is validated against items-mysql-monitor.dtd.

After defining the custom data collection, it is available to select in the Data Item drop-down menu on the Variable Assignment frame of the new Advisor page.

The following sections describe this process in detail.

### 22.3.1 Custom.xml

The following XML shows the structure of a custom data collection:

```xml
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE classes SYSTEM "items-mysql-monitor.dtd">
<classes>
  <class>
    <namespace>NameSpace</namespace>
    <classname>ClassName</classname>
    <precondition><![CDATA[Add Precondition Query Here]]></precondition>
    <query><![CDATA[Add Main Query Here]]></query>
    <attributes
      <attribute name="AttributeName1"/>
      <attribute name="AttributeName2"/>
    </attributes>
  </class>
</classes>
```

**Table 22.2 Custom Data Collection Class Elements**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>classes</td>
<td>Container element for all defined classes.</td>
</tr>
<tr>
<td>class</td>
<td>Container element for the definition of the collection.</td>
</tr>
<tr>
<td>namespace</td>
<td>Logical grouping for the new data collection item.</td>
</tr>
<tr>
<td>classname</td>
<td>Name of the custom data collection. Do not use spaces or special characters in this element.</td>
</tr>
<tr>
<td>precondition</td>
<td>(Optional) Query which checks some conditions. If the query returns true, the main query is executed. For example, the precondition query can be</td>
</tr>
</tbody>
</table>
Queries

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>used to check the version of the MySQL server. See Section 22.3.2.1, “Precondition Queries” for more information.</td>
<td></td>
</tr>
<tr>
<td>query</td>
<td>The main query. For more information, see Section 22.3.2.2, “Main Queries”</td>
</tr>
<tr>
<td>attributes</td>
<td>Enables you to label the types of data returned by the query. Possible types are: STRING, INTEGER, and FLOAT. This information is required by the advisor receiving the data. It is also possible to define one or more attributes as counters. See Section 22.3.3, “Data Collection Attributes” for more information.</td>
</tr>
</tbody>
</table>

The values in the namespace and classname elements are used as the first two elements of the name.

22.3.2 Queries

This section describes the precondition and main queries used to create custom data collections.

22.3.2.1 Precondition Queries

This section describes the optional precondition queries. Precondition queries determine that specific conditions are true before executing the main query. For example, they are used in the default advisors to check the MySQL server version, because some main queries cannot be executed on older versions of the server. The following is an example of a precondition query which checks the version of the MySQL server:

```
<precondition>
<![CDATA[SELECT @@version NOT LIKE '5.0%' AND @@version NOT LIKE '5.1%']]> </precondition>
```

If the server version is higher than 5.1, the precondition returns true and the main query is executed.
If the MySQL server is version 5.0.x or 5.1.x, the precondition returns false and the main query is not executed.

22.3.2.2 Main Queries

The main queries enable you to retrieve data from the monitored server.

When defining queries, the following restrictions apply:

- The query must be defined within a `<![CDATA[ ]]>` container. For example: `<![CDATA[SELECT X FROM Y AS FOO]]>`. Do not enter any characters between CDATA and the following [, nor between the [ and the start of the query. The same rule applies to the closing ]].

- Only SELECT statements are possible. It is not possible to use INSERT, UPDATE, DELETE, and so on.

- It is not possible to define more than one query per class.

- The agent must have sufficient rights to run the query.

- Do not define queries which take longer to run than the schedule defined on the advisor. For example, if the query takes 2 minutes to run, but the advisor-defined schedule requires the query to run every 1 minute, no results are returned. To avoid this, test your query thoroughly on the monitored server. If the custom data collection is deployed on multiple agents, it must be tested on each monitored server and the schedule modified accordingly.

- The query can return only one row, except if the result type `CLASS_TYPE_1STCOL_ATTRIBUTES` is used. See Section 22.3.3.2, “Returning Multiple Rows” for more information.
For each value retrieved from the server, you must assign a name. That is, you must use the following format, where NAME is the name applied to the data collection:

```
SELECT X AS NAME FROM Y
```

The items are displayed in the Data Item drop-down menu on the Variable Assignment frame of the new Advisor page. They take the following format: namespace:classname:name. For example, mysql:status:open_files_limit.

**Note**

The examples used in this section are taken from the default advisors delivered with your MySQL Enterprise Monitor installation.

The following example is used by the Server Has Anonymous Accounts advisor:

```
<class>
  <namespace>mysql</namespace>
  <classname>anonymous_user</classname>
  <query><![CDATA[SELECT COUNT(*) AS user_count FROM mysql.user WHERE user='']]></query>
</class>
```

In this advisor, the variable %user_count% is mapped to the Data Item mysql:anonymous_user:user_count defined in the query.

### 22.3.2.3 Wiki Formatting in Queries

It is possible to format the query result with wiki markup. This enables you to display information from the query directly in the event generated by the advisor.

The following example is taken from the data collection used by the Server Has Accounts Without A Password advisor:

```
<query><![CDATA[SELECT GROUP_CONCAT('\\n* ', '"',user,'"@"',host,'"' ORDER BY user, host) as user FROM mysql.user WHERE password='!'50507 AND (plugin = '"' OR plugin IS NULL OR plugin = 'mysql_native_password') OR (plugin = 'sha256_password' AND authentication_string = '')]]></query>
```

The wiki markup formats the user and host into information readily displayed in the Events page of MySQL Enterprise Monitor User Interface. This example lists the user name and host for all accounts without a defined password.

See Section 22.2.7, “Wiki Format” for more information on the supported wiki markup.

### 22.3.3 Data Collection Attributes

To properly evaluate the data returned by the data collection, assign attributes to the returned values.

Attributes are defined using the following format:

```
<attributes>
  <attribute name="AttributeName1" counter="true" type="INTEGER"/>
  <attribute name="AttributeName2" counter="false" type="STRING"/>
</attributes>
```

**Table 22.3 Attribute Elements**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the attribute defined in the AS clause of the query.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>counter</td>
<td>Whether the attribute is a counter type.</td>
</tr>
<tr>
<td>• true: the attribute is a counter type.</td>
<td></td>
</tr>
<tr>
<td>• false: the attribute is not a counter type.</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The attribute value type. Possible values are INTEGER, STRING or FLOAT.</td>
</tr>
</tbody>
</table>

**Important**

If an attribute type is incorrectly defined in the attribute definition, such as INTEGER instead of STRING, it is not possible to change the value in the custom.xml after the agent has started. This is because it is not possible for the agent to alter attribute types after they are defined. Attempting to change it in that manner results in an **InvalidValueForTypeException** error. To correct this, you must stop the agent, edit the type definition, rename the attribute, and restart the agent.

### 22.3.3.1 Default Values

If all the attributes are of the same type, it is not necessary to define the types for each attribute. Instead, define a default element at the beginning of the attribute list. In the following example, the default element assigns the same counter and type to each attribute:

```xml
<attributes>
  <default counter="true" type="INTEGER"/>
  <attribute name="bytes_read"/>
  <attribute name="bytes_written"/>
</attributes>
```

It is possible to override the default setting by assigning a counter, type, or both to the attribute definition. For example:

```xml
<attributes>
  <default counter="true" type="INTEGER"/>
  <attribute name="total_wait_time_ms"/>
  <attribute name="total_statements"/>
  <attribute name="max_wait_time_ms" counter="false"/>
  <attribute name="total_errors"/>
  <attribute name="total_warnings"/>
  <attribute name="total_rows_returned"/>
  <attribute name="total_lock_time_ms"/>
</attributes>
```

### 22.3.3.2 Returning Multiple Rows

It is possible to return more than one row, using the result type **CLASS_TYPE_1STCOL_ATTRIBUTES**. This result type enables the return of a two-column result set as key-value pair. Unlike the default attributes, which are taken from the column name, the key is the attribute name and the value is the attribute value.

**Important**

The key value must be unique across the result set.

The following example shows how a 2-column result set is returned and formatted by the **resulttype** element:
22.4 Event Notification Blackout Periods

During maintenance periods for database servers, you can suspend Event Handlers. During such a blackout period, Event Handlers are suspended. Agents continue to collect data, data is stored in the repository, and events are generated and displayed. Notifications, such as SNMP traps, emails and so on, are not generated.

To enable a blackout period for an individual instance, you can use the context menu on the MySQL Instances page. Open the instance menu and select Enable Event Handler Blackout. The instance name is greyed out to indicate the presence of an active blackout. No Event Handlers are triggered for the selected instance for the duration of the blackout period.

You can also enable a blackout period by entering the following URL into the address bar of your browser, substituting the appropriate host name, port and server name:

```
https://HostName:18443/rest?command=blackout&server_name=ServerName:3306&blackout_state=true
```

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

When the HTTP authentication dialog box requesting your MySQL Enterprise Monitor User Interface user name and password opens, specify the credentials for the Manager user. Use the ID and password you specified when you initially logged in to the Monitor UI.

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:

```
https://localhost:18443/rest?command=blackout&group_name=Finance&blackout_state=true
```

When the HTTP authentication dialog box opens, enter the administrator's credentials.

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

To reactivate the blacked-out server or server group, use the appropriate URL and query string, changing the blackout_state=true name/value pair to blackout_state=false. Again, this must be done by a user with administrative privileges.

Note

Restarting MySQL Enterprise Monitor does not reactivate a blacked out server.
22.4.1 Scripting Blackouts

You can write a script to black out a server, rather than opening a web browser and typing entries into the address bar. This section documents a sample blackout script that can be run from the command line.

Create the following file and save it as `blackout.pl`.

```perl
#!/usr/bin/perl
use LWP 5.64;

# USAGE: blackout.pl servicemanager:18443 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], 'MEM', $ARGV[1], $ARGV[2] );

my $url = URI->new('https://'.$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 $response->status_line . "\n";
}
if ($response->content =~ /UserUnauthorizedException/ ||
    $response->content =~ /ServerDoesNotExistException/) {
    die $response->content;
}
```

**Note**

Windows users can omit the initial `#!` line.

On Unix systems, use the `chmod +x blackout.pl` command to make the file executable.

At the command line, enter `blackout.pl servicemanager:18443 admin password servername:3306 true`.

Check the `configuration_report.txt` file for the host name and port to use. Specify the correct port for the Tomcat server. Specify the server to black out using the name that appears in the Server Tree, including the colon and port number as shown in the preceding example. Specify the name of a user who is a "manager". A user with "dba" rights cannot black out a server, and the script does not display any error in this case.

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

Restarting MySQL Enterprise Monitor does not reactivate a blacked out server.
Part IV Using the Query Analyzer
# Table of Contents

23 Using the Query Analyzer ................................................................. 251
  23.1 Providing Query Analyzer Data ................................................... 251
    23.1.1 Using the MySQL Performance Schema .................................. 252
  23.2 Query Response Time index (QRTi) .............................................. 255
  23.3 Query Analyzer User Interface ..................................................... 256
    23.3.1 Getting Detailed Query Information ....................................... 258
    23.3.2 Using Graphs to Identify Queries ......................................... 261
    23.3.3 Filtering Query Analyzer Data .............................................. 261
    23.3.4 Query Analyzer Settings ...................................................... 263
    23.3.5 Exporting Query Information ................................................. 264
Chapter 23 Using the Query Analyzer

Table of Contents

23.1 Providing Query Analyzer Data ................................................................. 251
   23.1.1 Using the MySQL Performance Schema ............................................. 252
23.2 Query Response Time index (QRTi) ......................................................... 255
23.3 Query Analyzer User Interface ............................................................... 256
   23.3.1 Getting Detailed Query Information ................................................. 258
   23.3.2 Using Graphs to Identify Queries ..................................................... 261
   23.3.3 Filtering Query Analyzer Data ......................................................... 261
   23.3.4 Query Analyzer Settings ................................................................. 263
   23.3.5 Exporting Query Information .......................................................... 264

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

Query Analyzer collects information about SQL statements that a MySQL client application sends to the MySQL server. There are different methods that the Query Analyzer can receive this information, which are:

• Using the Performance Schema statement digests with MySQL Server 5.6.14 and above, data can be gathered directly from MySQL Server without additional configuration, using a MySQL Enterprise Monitor Agent.

• The client application can route its database requests through the Proxy and Aggregator. The Proxy routes the client’s query to both the MySQL instance and the Aggregator. The Aggregator normalizes the queries and transmits them to the Service Manager.

• Install a MySQL Enterprise Monitor Plugin for a Connector that sends the information directly to MySQL Enterprise Service Manager.

Once your MySQL client application is configured to communicate via the MySQL Enterprise Monitor Agent, queries are monitored and the normalized queries are sent to the MySQL Enterprise Monitor Agent.

For the different ways to enable Query Analysis, see Section 23.1, “Providing Query Analyzer Data”. For the user interface of the Query Analyzer, see Section 13.2, “The Query Analyzer” and Section 23.3, “Query Analyzer User Interface”.

Once the data is collected, you view and monitor the queries, check the execution statistics, and filter and drill down on the information. By comparing the queries to the server graphs, you can correlate query execution with server status. For more information on viewing, filtering and reporting on the Query Analyzer data, see Section 23.3, “Query Analyzer User Interface”.

Note

When MySQL Enterprise Monitor is not accessible from a Connector/J or Connector/NET query analyzer plugin, the application performance is not impacted. Over time, the plugin determines that a backlog of reportable data exists, and fall back to consolidating it over longer ranges of time. But if more than 1,000 canonical queries are being used by the application (an unlikely scenario), data is dropped.

23.1 Providing Query Analyzer Data
The MySQL Query Analyzer can be fed information from a number of different sources. The provider supplies the statistical information about the queries, execution times, result counts and other data to display and analyze on the Query Analyzer page.

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

- Using the Performance Schema statement digests with MySQL Server 5.6.14 and above, data can be gathered directly from MySQL Server without additional configuration.

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

Using this method requires a connector that is capable of collecting and sending the query statistical data directly to MySQL Enterprise Service Manager. The connectors collect the basic query statistics, such as the execution time for each query, and the row counts, and provide this information to MySQL Enterprise Service Manager for analysis.

Note
This implementation type does not require the proxy component.

- Using the MySQL Enterprise Monitor Proxy and Aggregator. For more information, see Chapter 11, Proxy and Aggregator Installation.

Important
If you are using the MySQL Enterprise Monitor Proxy and Aggregator to collect query performance data, you must disable the statements_digest consumer in performance_schema.setup_consumers.

23.1.1 Using the MySQL Performance Schema

As of MySQL Enterprise Monitor 3.0.0, Query Analyzer data is automatically collected and displayed by simply monitoring MySQL Server 5.6.14 or greater, and without any additional plugins required. This ability comes from the Performance Schema Statement Digests feature (Performance Schema Statement Digests) that was added in MySQL 5.6. If you are using an earlier version of MySQL Server (5.6.13 or below), then you can continue to use a Connector Plugin or MySQL Proxy to provide performance information to the Query Analyzer.

Note
MySQL server versions prior to MySQL 5.6.14 are disabled due to a crashing bug within Statement Digests that could be triggered by collecting the data from the Agent.

Collecting Query Analyzer data from Performance Schema, rather than at the wire protocol (which is how the other sources of Query Analyzer data work) provides data about what the statements do to generate their result sets that other sources cannot provide:

- Table Lock time
- How many rows were examined versus returned
- How many temporary tables were created, and whether any were created on disk
- Whether range scans were done, and in what form they were done
- Whether sorting happened, how many rows were sorted, and what form the sort took

There is also information not available when operating in this mode that is provided to the Query Analyzer when using Connector Plugins and MySQL Proxy:
• Stack trace of where the statement originated from on the application side (Connector Plugins only)
• Histograms of response times
• Standard deviation of response times

When enabled (which is the default), the MySQL Enterprise Monitor Agent polls the
`performance_schema.events_statements_summary_by_digest` table (every minute, by
default) and continually compute the deltas for each of the normalized statements that are exposed
during the snapshot window. This is dependent on the Performance Schema setup having the
"statements_digest" consumer enabled within `performance_schema.setup_consumers`, which is
enabled by default in MySQL 5.6:

```sql
mysql> SELECT * FROM performance_schema.setup_consumers WHERE name = 'statements_digest';
+-------------------+---------+
| NAME              | ENABLED |
| statements_digest | YES     |
+-------------------+---------+
```

If this is not enabled, then enable it with:

```sql
UPDATE performance_schema.setup_consumers SET enabled = 'YES' WHERE name = 'statements_digest';
```

**Note**
The MySQL Enterprise Monitor Agent does not TRUNCATE the `performance_schema.events_statements_summary_by_digest` table each time it reads from it, as it is possible there may be other processes/tools consuming this data. Because of this, the "Max Latency" statistic that is reported per a normalized statement within Query Analyzer is actually the maximum since either the MySQL Server started, or since a TRUNCATE TABLE `performance_schema.events_statements_summary_by_digest` was executed. This differs from the MySQL Proxy or Connector Plugins, which report the maximum run time per the aggregated snapshot period.

**Note**
The maximum space available for digest computation is 1024 bytes by default; queries exceeding this length are truncated.

As of MySQL 5.7.8, and later, and 5.6.26, and later, this value can be changed at server startup by setting the `performance_schema_max_digest_length` system variable. In MySQL 5.6.24, 5.6.24, 5.7.6, and 5.7.7, use `max_digest_length` instead. For MySQL 5.7 versions prior to 5.7.6, the value cannot be changed. Nor can it be changed for MySQL 5.6 versions prior to 5.6.24.

The `performance_schema.events_statements_summary_by_digest` table is a sized table in memory within the Performance Schema, and its size is auto-configured. To check the current size:

```sql
mysql> SHOW GLOBAL VARIABLES LIKE 'performance_schema_digests_size';
+-------------------+-------+
| Variable_name     | Value |
| performance_schema_digests_size | 5000  |
```
If your application executes more than this number of normalized statements, then it is possible that you may begin losing some statement instrumentation. You can monitor this situation with the `Performance_schema_digest_lost` system variable:

```
mysql> SHOW GLOBAL STATUS LIKE 'Performance_schema_digest_lost';
+------------------------+-------+
| Variable_name           | Value |
|------------------------+-------+
| Performance_schema_digest_lost | 0     |
```

If you detect that this counter variable is growing, consider increasing the `performance_schema_digests_size` system variable. It is also possible that your statement profile has changed over time, and that you are now executing different statements than were originally tracked (this is especially possible in very long running instances). In this case, you can simply TRUNCATE TABLE `performance_schema.events_statements_summary_by_digest`, and the Query Analyzer collection automatically starts again.

When the "Example Query" feature is enabled, Query Analyzer attempts to get an example of the longest running statement during the snapshot interval by doing a LEFT JOIN with a groupwise-max on the `performance_schema.events_statements_summary_by_digest` table to the `performance_schema.events_statements_history_long` table. Using this method does not guarantee that an example statement is always provided because, by default, the `events_statements_history_long` table is a ring buffer of the last 1000 statements executed. This too differs from the Connector Plugin and MySQL Proxy sources, which always provide an example per normalized statement, per snapshot, when enabled. We collect in this way with Performance Schema to minimize load on the monitored instance rather than polling the `performance_schema.events_statements_history_long` table at too high a frequently to try and gather statistics.

---

**Note**

A small subset (approximately 2MB of data) of the snapshot of known prior values is retained in-memory, and the rest is spooled to disk. The spool is stored in `$MYSQL_AGENT_HOME/spool/queryAnalysis`.

The "Example Query" feature requires that the `events_statements_history_long` table is enabled within `performance_schema.setup_consumers` (this is disabled by default within MySQL 5.6):

```
mysql> SELECT * FROM performance_schema.setup_consumers where name = 'events_statements_history_long';
+------------------------+---------+
| NAME                   | ENABLED |
|------------------------+---------+
| events_statements_history_long | NO      |
```

If this is not enabled, then enable it with:

```
UPDATE performance_schema.setup_consumers SET enabled = 'YES' WHERE name = 'events_statements_history_long';
```

When "Example Query" and "Example Explain" are enabled, the MySQL Enterprise Monitor Agent attempts to run an EXPLAIN for each example statement that is discovered and ran for longer than the...
"Auto-Explain Threshold". Due to the way that Performance Schema exposes normalized statements, truncating any normalized statement that is longer than 1024 bytes due to memory concerns within the MySQL Server means it is possible that an EXPLAIN may fail because the truncated statements do not parse correctly when running the EXPLAIN.

23.2 Query Response Time index (QRTi)

QRTi stands for "Query Response Time index". It is a "quality of service" measurement for each query, and it uses the Apdex formula for that calculation: Apdex on Wikipedia.

How QRTi is Defined

The three measurement conditions are "optimum", "acceptable", and "unacceptable", which are defined as:

<table>
<thead>
<tr>
<th>Type</th>
<th>Default time values</th>
<th>Assigned value</th>
<th>Description</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimum</td>
<td>100ms</td>
<td>1.00 (100%)</td>
<td>The optimal time frame</td>
<td>Green</td>
</tr>
<tr>
<td>Acceptable</td>
<td>4 * Optimum -- 100ms to 400ms</td>
<td>0.50 (50%)</td>
<td>An acceptable time frame</td>
<td>Yellow</td>
</tr>
<tr>
<td>Unacceptable</td>
<td>Exceeds Acceptable -- greater than 400ms</td>
<td>0.00 (0%)</td>
<td>An unacceptable time frame</td>
<td>Red</td>
</tr>
</tbody>
</table>

An example calculation

From there, we calculate an average to determine the final QRTi value. For Example, if there are 100 executions of the digested/canonical query, where 60 finished below 100ms (the optimal time frame), 30 between 100ms and 400ms (the acceptable time frame), and the remaining 10 took longer than 400ms (unacceptable time), then the QRTi score is:

\[
\frac{(60 + (30 / 2) + (10*0))}{100} = 0.75. 
\]

Reading QRTi Values

The queries listed on the Query Analyzer page also have a color coded pie chart representing a breakdown of the values used in the QRTi calculation (green representing the optimal time frame, yellow the acceptable time frame, and red the unacceptable). You can mouse over the pie chart itself to see the total number of query executions that fell within each category, as well as the percentage of query executions that fell within that group.

So when doing query optimization, you want to start with the ones that have a QRTi visual pie chart that is 100% red, which means that they also have an actual QRTi value of 0. This means that "all" executions of that query took longer than the acceptable time frame (400ms by default). You can then click on the query to get more information, such as the maximum and average query times, the average number of rows examined, the average lock wait time, examine a sample query, look at an example EXPLAIN plan, see if full table scans were done, examine index usage, etc.

You can then work your way up from the queries with a QRTi value of 0, towards those that have a value of 1 (1 meaning that all instances of the query executed within the optimal time frame). Once you get to the point that you no longer have any queries with a QRTi value of less than 1, then you can go into the Query Analysis Reporting Advisor configuration, and adjust the QRTi Threshold (the target time) down, say to 50ms, and start the process all over again.
23.3 Query Analyzer User Interface

To analyze the queries captured by the agent/proxy, change to the Query Analyzer tab. The following figure shows an example of the table on that page:

Figure 23.1 MySQL Enterprise Monitor User Interface: Query Analyzer

The main Query Analyzer table provides the summary information for all of the queries executed via the agent/proxy. The table tracks all the queries submitted to the server via the agent/proxy. The table shows a maximum of 100 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 23.3.3, "Filtering Query Analyzer Data". To export the query information as a .csv file, see Section 23.3.5, "Exporting Query Information".

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 shows 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 is configured to show queries within the last 30 minutes (Interval), then only queries executed during that time are displayed, and the corresponding statistics, such as execution times, rows returned and bytes returned, reflect that 30 minute timespan.

- **Query**: The normalized version of the query. Normalization removes the query-specific data so that different queries with different data parameters are identified as the same basic query.

  The information is shown as one query per row. Each query row is expandable, and can be expanded to show the execution times for individual servers for that query.

- **Database**: The default database in use at the time of the query. The database name might be blank, or might not match the database used within the query, if you used a qualified table name (for
example, `select ... from db_name.table_name` or if you issued a `USE` statement to switch databases after connecting.

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

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

- **QRTi**: Lists the Query Response Time index of the query. For more information, see Section 23.2, “Query Response Time index (QRTi)”.

- **Latency (hh:mm:ss.ms)**: The execution time for all the matching queries. This is the time, for every invocation of the corresponding query, as calculated by comparing the time when the query was submitted and when the results were returned by the server. Times are expressed in HH:MM:SS.MS (hours, minutes, seconds, and milliseconds).

The **Exec Time** column is further subdivided into the following columns:

- **Total**: The cumulative execution time for all the executions of this query.
- **Max**: The maximum execution time for an execution of this query.
- **Avg**: The average execution time for the execution of this query.
- **Locks**: the time spent waiting for table locks caused by the query.
- **Average History graph (Avg History)**: graphs the average execution time.

When looking at the information provided in this query, compare 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 could indicate an issue that needs to be investigated.

- **Rows**: The rows returned by the query. The column is sub-divided into the following columns:

  - **Total**: The sum total number of rows returned by all executions of the query.
  - **Max**: The maximum number of rows returned by a single execution of the query.
  - **Avg**: The average number of rows returned by all executions of the query.

- **Bytes**: The number of bytes returned by each query. The column is sub-divided into the following columns:

  - **Total**: The sum total bytes returned by all executions of the query.
  - **Max**: The maximum number of bytes returned by a single execution of the query.
  - **Avg**: The average number of bytes returned by all executions of the query.

- **First Seen**: The date and time the normalized version of this query was first seen, which might be earlier than the period specified by the filter.

You can sort the list of queries by clicking the column name. The direction of the sort (highest to lowest, or lowest to highest) is indicated by a triangle next to the currently selected column. The default is to sort the list of queries by the **Latency:Total** time.

To help you and locate queries you can filter the list of queries using a variety of criteria. For more information on the filtering methods available, see Section 23.3.3, “Filtering Query Analyzer Data”.
23.3.1 Getting Detailed Query Information

Click on an individual query to see more detailed information about the individual query in a pop-up window, as shown in Figure 23.2, “MySQL Enterprise Monitor User Interface: Canonical Query Tab for a Query”. The available tabs within this window depend on whether you have configured the more detailed query information. By default, you see the Canonical Query view.

You can also view Example Query, which provides more detailed data about a specific query, including the data and parameters submitted. You can also enable Explain Query, which lets you remotely execute an EXPLAIN statement with the specified query and view the resulting information. Finally, you can view any graph data produced during the execution of the query by using the Graphs tab.

• The Canonical Query tab:

![Figure 23.2 MySQL Enterprise Monitor User Interface: Canonical Query Tab for a Query](image)

The canonical view for a query provides three different views of the query, which can be changed using the links under the Canonical Form section. The truncated version is a shortened version of the query. The full version of the query is the entire query statement. Normalization removes the constants from the individual queries so that queries following the same logical structure are identified as the same basic query.

**Note**

The "full" version of statements provided by the digested Performance Schema may be truncated, as the Performance Schema statement digest may truncate the statement due to memory constraints.

In addition to the summary information given in the table, the Execution Time Statistics section provides you with more detailed execution time statistics, including the minimum time, maximum time, average time, total time and the standard deviation. The standard deviation lets you determine whether a particular invocation of a query is outside the normal distribution of times for the given query.
The **Row Statistics** provide more detailed contents on the maximum, minimum, average, total, and standard deviation for the number of rows returned by the query, and the total size and maximum size of the data returned. The time period for the total and average figures is shown under the **Time Span** header.

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

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

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

- **The Example Query** tab:

  **Figure 23.3 MySQL Enterprise Monitor User Interface: Example Query Tab for a Query**

  ![](image)

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

  In addition to the full query, with data, that was executed, the tab shows the execution time, data, user, thread ID, client host and execution host for the given query.

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

- **The Explain Query** tab:
The **Explain Query** tab lets you view the output from running the query with the `EXPLAIN` prefix. For more information, see [EXPLAIN Syntax](#).

---

**Important**

Explain queries are generated for query data supplied by the MySQL Enterprise Monitor Proxy and Aggregator, Connector/J plugin, and Performance Schema sources.

Explain is supported for all DML statements on MySQL 5.6.3 or higher. On earlier versions, only `SELECT` is supported.

---

**Note**

If the Query Analyzer is used with the MySQL Enterprise Monitor Proxy and Aggregator, EXPLAIN is not performed for any `SELECT` statement which uses `SQL_CALC_FOUND_ROWS`.

---

- **The Graphs tab:**

---

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

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

You can use the correlated graphs in two different ways:

- Drag and select an area on a graph as displayed within the Overview dashboard, or from the Graphs & Reports page of the MySQL Enterprise Monitor User Interface. You can drag and select any region on a displayed graph, click the QUAN icon, and it loads the selected range into the Query Analyzer page, displaying the corresponding zoomed graph, and the associated queries being executed during the selected period.

- Do the same, but select the graph from within the Query Analyzer page.

Figure 23.6 MySQL Enterprise Monitor User Interface: Correlated Graphs

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

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

To export the graph, see Chapter 16, Reports and Graphs.

23.3.3 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 filtering functionality is available in a simple format, supporting simple statement and timing based filtering, and an advanced option allowing you to filter by specific columns within the Query Analyzer table.
The basic filter options are:

- **Statement Text** and **Value** support text searching of the normalized query. For the search type you can specify either a basic text match (Contains), or a regular expression match (Regex). In addition to the basic text match, you can also search for a query that does not contain a particular string. For regular expression searches, you can specify whether the regular expression should match, or not match (negative regexp) the queries. Regular expressions are parsed using the standard MySQL REGEXP() function. For more information, see Regular Expressions.

  **Note**

  The search is performed against the canonical version of the query. You cannot search against specific text or values within the parameters of the query itself.

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

- **DB Name**: Limits the queries to those executed within a specific database. The database match is performed using the LIKE match from the MySQL database, hence you can use the % and _ characters to multiple and single character matches. For more information, see Pattern Matching.

- The **Time Range** menu selects whether the time selection for filtering should be based on the time interval (only queries recorded within the displayed time period are shown, using the Hours and Minutes pop-up), or whether the selection should be based on a time period (From/To), where you can select the time range to be displayed.

  Using the Interval mode shows queries within the given time period from the point the graph was updated. For example, if you select 30 minutes, then the queries shown were captured within the last 30 minutes. If you updated the display at 14:00, then the queries displayed would have been captured between 13:30 and 14:00. Using interval mode limits the timespan for the filter selection to a maximum of 23 hours and 59 minutes.

  Using the From/To time range lets you show queries between specific dates and times. Using this mode you can show only the queries received during a specific time span, and you can display the query history for a much longer time period, for as long as you have been recording query analysis information.

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

To use the advanced filtering techniques, click **show advanced**. This provides additional filters:

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

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

  To use the column filters, you must specify the **Column** that you want to filter on, the **Operator** to use when performing the comparison and the **Value** that you want to compare.
For example, to filter by showing all the queries that return more than 100 rows on average, set the Column to Average Rows, the Operator to >=, and the Value to 100.

- The View selection determines whether the information should be returned on a group basis, where an aggregate of the same query executed on all monitored servers is shown, or on a Server basis, where queries are summarized by individual server. If the latter option has been selected, the table includes an additional column showing the server.

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

When you have set your filter parameters, you can update the Query Analysis display by clicking the filter button. To reset the fields to the default settings click the reset button.

If you want to make the current filter options the default when viewing the Query Analyzer page, click the Save As Default button. The settings are saved for the current user only.

23.3.4 Query Analyzer Settings

There are a number of settings related to the Query Analyzer data. To configure the Query Analyzer, go to the Configuration, Advisors page and select the Query Analysis Advisor category. Then choose Edit Advisor Configuration from the context menu of the Query Analysis Reporting Advisor.

Figure 23.8 MySQL Enterprise Monitor User Interface: Query Analyzer Configuration

Like with any Advisor, this may be set globally, or for a group or particular MySQL server.

Thee configuration options are:

- **Enable Example Query** displays more information about individual queries. When enabled, queries and their data items (rather than the canonical form shown by default) are provided. Enabling this option could expose the full query statements and therefore could present a security issue.

  With the Example Query option enabled, an additional tab is available within the query summary details. For more information, see Section 23.3.1, “Getting Detailed Query Information”.

  If you enable Example Query, you can also 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 shows the full query and how the query was executed within the servers.

  Enabling this option might add overhead to the execution of your server, as the server runs an EXPLAIN statement each time it identifies a long running query. For more information, see Appendix A, MySQL Enterprise Monitor Frequently Asked Questions.

- **Auto-Explain Threshold**: EXPLAINs are generated for queries with a runtime above this threshold. (Format: hh:mm:ss.msec)
• **QRTi Threshold**: Optimal time for response time index. For more information about QRTi, see Section 23.2, "Query Response Time index (QRTi)".

You can also define a schedule for the advisor's data collection. For more information, see Section 19.4, “Advisor Schedules”

---

**Important**

If this advisor is disabled, Query Analysis data is no longer collected from the monitored instances. The Query Analyzer continues displaying data collected prior to the advisor being disabled.

If disabled, a message is displayed on the Query Analyzer page: "N of the selected servers do not have query analyzer enabled." where N is the number of servers.

---

### 23.3.5 Exporting Query Information

To get the text and details of the queries displayed on the Query Analyzer page, click one of the icons to the right of the **Browse Queries** label in the separator bar. The query information is exported as comma-separated data in a **.csv** file, with fields corresponding to the columns shown in the Monitor UI, and the time of the export encoded in the filename in UTC format. The icon representing a single page exports the query information for the currently displayed page only. The icon representing a stack of pages exports the query information for all available pages of query information.

Within the **Query Analyzer** popup, you can also export information about each data group as a **.csv** text file or a **.png** image file, using icons next to the labels on the left side. The output file is named according to the pattern **Statement_Report_Summary_%server/group%_%creationtimestamp%.csv**. The data exported by these icons includes:

- Execution Time: Count
- Executions: Exec Time, Max Exec Time, Min Exec Time, Average Exec Time
- Rows: Rows, Max Rows, Min Rows, Average Rows
- Kilobytes: Bytes, Max Bytes, Average Bytes

**Note**

Microsoft Excel users on Windows users can import the **.csv** file as a spreadsheet. If the file contains English text, typically you can double-click it to open in Excel. If the file contains localized Japanese text, you must use the **File > Open** menu choice within Excel to open the file.

In the Safari browser, exported files containing localized data might contain %NN character sequences in their names, due to browser issues with UTF-8 and Base64 encodings.
Part V Appendices
# Table of Contents

A MySQL Enterprise Monitor Frequently Asked Questions ................................................. 269
B Managing the Inventory ........................................................................................................ 275
  B.1 The Inventory Page ............................................................................................................ 275
  B.2 Using the Inventory Page .................................................................................................. 275
C MySQL Enterprise Monitor General Reference .................................................................... 277
  C.1 MySQL Enterprise Monitor Limitations ............................................................................ 277
  C.2 Supported Browsers .......................................................................................................... 277
  C.3 Backing up MySQL Enterprise Service Manager ............................................................. 278
  C.4 Regular MySQL Enterprise Monitor Maintenance ............................................................ 279
D MySQL Enterprise Monitor Component Reference ............................................................. 281
  D.1 MySQL Enterprise Service Manager Reference ............................................................... 281
    D.1.1 Log Files for the MySQL Enterprise Service Manager .............................................. 281
    D.1.2 The Management Information Base (MIB) File ......................................................... 281
    D.1.3 The config.properties file ........................................................................................ 282
    D.1.4 Using The Service Monitor Repository Configuration Script .................................... 286
  D.2 MySQL Enterprise Monitor Agent Reference ................................................................. 286
    D.2.1 MySQL Enterprise Monitor Agent Configurable Options ......................................... 286
    D.2.2 Agent Log Files .......................................................................................................... 286
    D.2.3 Using The Agent Configuration Script ........................................................................ 287
E Error codes ............................................................................................................................ 289
F Files associated with configuring, debugging, and using MySQL Enterprise Monitor ............ 299
G Data Collection Items ............................................................................................................ 301
H MySQL Enterprise Monitor Release Notes ........................................................................... 367
Appendix A MySQL Enterprise Monitor Frequently Asked Questions

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at http://www.mysql.com/products/.

FAQ Categories

- General Usage
- MySQL Monitor
- MySQL Query Analyzer

General Usage Questions

- A.1: How do I find Ignored MySQL Instances? And how to I show them again?
- A.2: In 2.3, the agent-mgmt-hostname contained the string "heartbeat" as the URLs path. Did this change?
- A.3: How do I change the name of a server?
- A.4: Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?
- A.5: Why do some rules appear to have a Severity of Unknown?
- A.6: Can I run MySQL Enterprise Service Manager on machine with other applications running?
- A.7: Why does the file apache-tomcat/logs/tomcat.log show error messages saying This is very likely to create a memory leak.? Is that anything to be concerned about?
- A.8: Why does monitoring a MySQL instance with FEDERATED tables cause extra connections, and decreased performance?

Questions and Answers

A.1: How do I find Ignored MySQL Instances? And how to I show them again?

From the MySQL Instances page, open the Unmonitored Instances panel and enable the Ignored Instance filter parameter and execute the search. This lists the ignored MySQL Instances.

To change the status of an ignored MySQL Instance, choose Show Instance from the context-menu for a specific MySQL Instance, or check the ignored MySQL Instance(s) and click the Show Instances button.

A.2: In 2.3, the agent-mgmt-hostname contained the string "heartbeat" as the URLs path. Did this change?

Yes, this is no longer required and is ignored as of MySQL Enterprise Monitor 3.0.0.

A.3: How do I change the name of a server?

Open the MySQL Instances dashboard, and choose Edit Instance from the instance menu. Alternatively, toggle the checkbox for one instance and click Edit Instances.

Renaming the server in this way will override all other server naming, including changes to the agent configuration.
A.4: Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?

MySQL 5.1 or later is supported.

Analyzing Performance Schema results requires MySQL Server 5.6.14 and above.

A.5: Why do some rules appear to have a Severity of Unknown?

Due to timing issues, certain rules such as “32-Bit Binary Running on 64-Bit AMD Or Intel System” and “Key Buffer Size Greater Than 4 GB” do not evaluate correctly due to timing issues. This is a known issue that is expected to be resolved in future versions of MySQL Enterprise Monitor.

A.6: Can I run MySQL Enterprise Service Manager on machine with other applications running?

You can, but Oracle recommends running your MySQL Enterprise Service Manager on a dedicated machine, especially when monitoring many agents.

A.7: Why does the file apache-tomcat/logs/tomcat.log show error messages saying This is very likely to create a memory leak.? Is that anything to be concerned about?

This message is sometimes produced by underlying components of the web stack on web application reload or shutdown, and is not a cause for concern. It is not practical to shut off these spurious messages within Tomcat.

A.8: Why does monitoring a MySQL instance with FEDERATED tables cause extra connections, and decreased performance?

When the agent starts, it executes a discovery process that performs a number of INFORMATION_SCHEMA queries that gather table information for rules. These INFORMATION_SCHEMA queries can be costly on instances with many tables, particularly with large numbers of FEDERATED tables to another instance, as each table has a new session opened for it on the target machine.

MySQL Monitor

Questions

• A.1: What is MySQL Enterprise Monitor?
• A.2: What are the features and related benefits of the MySQL Enterprise Monitor?
• A.3: What are the immediate benefits of implementing the MySQL Enterprise Monitor?
• A.4: What are the long-term benefits of the MySQL Enterprise Monitor?
• A.5: How is the MySQL Enterprise Monitor installed and deployed?
• A.6: How is the Enterprise Monitor web application architected?
• A.7: What makes MySQL Enterprise unique?
• A.8: What versions of MySQL are supported by the MySQL Enterprise Monitor?
• A.9: What operating system platforms are supported by the MySQL Enterprise Monitor?
• A.10: What are the MySQL Enterprise Monitor Advisors?
• A.11: How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?

Questions and Answers

A.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 Oracle Software Delivery Cloud web site and is deployed within the safety of the customer data center.

A.2: What are the features and related benefits of the MySQL Enterprise Monitor?

The MySQL Enterprise Monitor is like having a “Virtual DBA Assistant” at your side to recommend best practices to eliminate security vulnerabilities, improve replication, and optimize performance. For the complete features and benefits, visit the http://www.mysql.com/products/enterprise/monitor-features.html.

A.3: 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.

A.4: 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.

A.5: 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.

A.6: How is the Enterprise Monitor web application architected?

The Enterprise Monitor web application comprises three components:

• Monitor Agent: A lightweight Java program that is installed on each of the monitored hosts. 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 and host. The Monitor Agent is the only component within the application that touches or connects to the MySQL Server. It reports the data it collects via XML over HTTP to the centralized Service Manager.

• Service Manager: The main server of the application. The Service Manager manages and stores the data collections that come in from each monitor agent. It analyzes these collections using MySQL provided best practice Advisor rules to determine the health, security, availability and performance of each of the monitored MySQL Servers. The Service Manager also provides the content for the Enterprise User Interface 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.

A.7: 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.

A.8: What versions of MySQL are supported by the MySQL Enterprise Monitor?

The MySQL Enterprise Monitor supports MySQL versions 5.1 and above.

A.9: 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 and Windows Server Editions, and Solaris. The Monitor Agent supports any platform supported by the MySQL Enterprise server. For the complete list of MySQL Enterprise supported operating systems and CPUs, visit MySQL Supported Platforms and select MySQL Enterprise Monitor.

A.10: What are the MySQL Enterprise Monitor Advisors?

Advisors filter and evaluate the information broadcast by the Monitoring Agents and present it to the Events page when defined thresholds are breached. They also present advice on how to correct the issue. There are more than 200 Advisors, all of which are enabled by default. Thresholds are the predefined limits for Advisors. If the monitored data breaches the defined threshold, an event is generated and displayed on the Events page. Advisor thresholds use a variety of different value types, depending on the monitored value. Some use percentages, such as percentage of maximum number of connections. Others use timed durations, such as the average statement execution time. It is also possible to check if specific configuration elements are present or correct.

A.11: How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?

Customers receive email notifications of new and updated MySQL Enterprise Monitor versions. Also, the What’s New section of MySQL Enterprise Monitor, if enabled, contains new product announcements.

MySQL Query Analyzer

Questions

• A.1: What is the MySQL Query Analyzer?
• A.2: How is the MySQL Query Analyzer installed and enabled?
• A.3: What are the main features and benefits of the MySQL Query Analyzer?
• A.4: What are the typical use cases of the MySQL Query Analyzer?
• A.5: What makes the MySQL Query Analyzer unique?
• A.6: How can I get the MySQL Query Analyzer?
• A.7: Does Query Analyzer work with MySQL Cluster?
• A.8: Does Query Analyzer enable me to monitor the disk reads and writes during a query?
• A.9: Does Query Analyzer handler prepared statements?
• A.10: How much degradation in performance does mysql-proxy introduce?
• A.11: Will the Query Analyzer work without any special setup?

Questions and Answers

A.1: What is the MySQL Query Analyzer?
The MySQL Query Analyzer allows DBAs, developers and system administrators to improve application performance by collecting, monitoring, and analyzing queries as they run on their MySQL servers. http://www.mysql.com/products/enterprise/query.html

A.2: How is the MySQL Query Analyzer installed and enabled?
See Section 23.1, “Providing Query Analyzer Data”.

A.3: What are the main features and benefits of the MySQL Query Analyzer?
For the complete features and benefits, see MySQL Enterprise Monitor Features and Benefits.

A.4: 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.

A.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.

A.6: How can I get the MySQL Query Analyzer?
The MySQL Query Analyzer is built into the MySQL Enterprise Monitor.

To experience the MySQL Enterprise Monitor for 30 days, visit the http://www.mysql.com/trials/

A.7: Does Query Analyzer work with MySQL Cluster?
Yes, providing that exact node is monitored with an agent and query analyzer has been enabled for that node. Note that you must be accessing your cluster data through a standard MySQL node for this to work.

A.8: Does Query Analyzer enable me to monitor the disk reads and writes during a query?
No, that information is not available to the query analyzer, but many Advisors and graphs do handle this information. An Agent monitors the host, which includes monitoring of the CPU, Disk, and Memory.

A.9: Does Query Analyzer handler prepared statements?
At this time, the query analyzer does not track server-side prepared statements. However the default configurations for most client-side libraries for MySQL don’t use them, they emulate them client-side, and those will be tracked by the query analyzer.

A.10: How much degradation in performance does mysql-proxy introduce?
At the very least it's equivalent to a network hop in latency. The degradation is directly related to your average query execution time. If your queries execute in microseconds (which can happen if served
from query cache) then the degradation will be higher, and noticeable. We’ve seen some applications that actually do work when they execute queries, the degradation is much less, and in some limited cases because of scheduling, the application actually has better throughput.

A.11: Will the Query Analyzer work without any special setup?

With MySQL Server 5.6.14 and greater, Query Analyzer data is automatically (by default) collected and displayed using the Performance Schema Statement Digests MySQL Server feature. If you are monitoring an earlier MySQL Server version, then you can continue to use alternative methods of providing query data to the Query Analyzer.

For information about the different methods of retrieving query data, see Section 23.1, “Providing Query Analyzer Data”.
Appendix B Managing the Inventory

Table of Contents

B.1 The Inventory Page ................................................................. 275
B.2 Using the Inventory Page ......................................................... 275

The Inventory pages enable you to view all currently monitored assets and delete assets which are no
longer monitored or no longer present. It is also useful for debugging problems with your setup. The
information in the Inventory page is read from the repository’s Inventory schema, where all information
about the current and historical assets is stored.

Historical assets are assets which were once monitored but are no longer used, such as servers
which used to host MySQL instances but were decommissioned, or repurposed. These persist in the
repository’s Inventory schema and are displayed in the MySQL Enterprise Monitor User Interface even
though they are no longer used.

Current assets are assets which are active and currently monitored.

B.1 The Inventory Page

The Inventory page cannot be accessed from the MySQL Enterprise Monitor User Interface. To open
the inventory page, you must edit the URL in the browser address bar. To open the Inventory page,
enter the following address in your browser:

https://ServiceManagerHost:PortNumber/v3/inventory

Where ServiceManagerHost is the address of your MySQL Enterprise Service Manager and
PortNumber is the port it listens on.

Enter the login details, if prompted to do so. The username and password are the same as those used
to log in to the MySQL Enterprise Monitor User Interface.

All Inventory

The All Inventory page displays all recorded assets, current and historical, grouped into categories.

For example, selecting agent.Agent opens a page listing all the agents stored in the inventory.
Selecting one of those agents, opens a page listing the details of that agent. Details such as the
homeDir, version, and so on.

All MySQL Servers

The All MySQL Servers page displays all current, monitored MySQL instances. A historical record
of instances is not kept. If a MySQL instance is deleted from the MySQL Enterprise Monitor User
Interface, it is deleted from the inventory and is not displayed in the All MySQL Servers inventory
page.

All Hosts

The All Hosts page displays all current and historical hosts. Clicking one of the host links opens a
page listing the details of that host. Details such as the number of CPUs, the filesystems and the
MySQL instances, if any, installed on that host.

B.2 Using the Inventory Page

The Inventory page enables you to view the details of all assets stored in the repository, and to delete
obsolete or unused assets.
Deleting Assets

MySQL Enterprise Monitor maintains a record, in the Inventory schema, of all assets detected. As a result, if the network topology changes frequently, the inventory and the MySQL Enterprise Monitor User Interface may contain many unused or obsolete assets. The Inventory page enables you to remove such assets, permanently.

Important
If a current asset, that is one which is actively monitored, is deleted, MySQL Enterprise Monitor rediscovers it as part of the monitoring process.

To delete an obsolete or unused asset, do the following:

1. Navigate to the asset's page.
2. Click the **Delete** button in the left-hand sidebar.
   
   A confirmation dialog is displayed, asking if you want to delete the asset.

3. Click **Yes** to delete the asset, **Cancel** to return to the asset page.

Important
To delete a host which is currently monitored, you must first, in the MySQL Enterprise Monitor User Interface, stop the monitoring Agent, delete the Agent and Instance, then delete the host using the Inventory page.
Appendix C MySQL Enterprise Monitor General Reference

Table of Contents

C.1 MySQL Enterprise Monitor Limitations ........................................................................................................ 277
C.2 Supported Browsers ....................................................................................................................................... 277
C.3 Backing up MySQL Enterprise Service Manager ......................................................................................... 278
C.4 Regular MySQL Enterprise Monitor Maintenance .......................................................................................... 279

This appendix contains reference information for MySQL Enterprise Monitor.

C.1 MySQL Enterprise Monitor Limitations

The following items are known limitations within MySQL Enterprise Monitor:

• The maximum size for each data item stored within the `mysql.inventory` table is 64 bytes.

• Fonts used for MySQL Enterprise Service Manager graphs might not be displayed properly if a required font is missing.

To specify a custom font, set a custom property within the embedded MySQL Server to a known working font. Connect to your embedded MySQL Server using the information provided by the `configuration_report.txt` file. Once connected, execute the following statement:

```
mysql> INSERT INTO map_entries VALUES (1,'Helvetica','graph.font');
```

Replacing `Helvetica` with the font to use.

The font used for graph rendering is selected as follows, in this order:

1. The user override value.
2. The MySQL Enterprise Service Manager default of Arial.
3. The graph engine default of SansSerif.

• You cannot monitor more than one MySQL Enterprise Monitor User Interface instance hosted on the same machine simultaneously within the same browser, even if each MySQL Enterprise Monitor User Interface uses a different port for access. The effect of this limitation can be more significant on certain browsers.

• When monitoring multiple MySQL instances with one MySQL Enterprise Monitor Agent, if any of the MySQL instances are down when the agent is started, the agent does not attempt to reconnect to the servers. This could lead to indications that MySQL instances were down when in fact they were running and available. To avoid this problem, ensure all of the monitored instances are available before starting the agent.

• The query length available for Query Analysis is limited, depending on the source. For example, a query that exceeds the limit is truncated and only displays part of the query in the "Example" section of the query analysis report.

With 3.0 Agents: When using the Performance Schema as a source, queries are limited to the digested Performance Schema limit of 1,024 characters.

With 3.0 Agents: When not using the Performance Schema as a source, the limits are set by the source of the data, if there is a limit at all.

C.2 Supported Browsers
The following browser versions are recommended for use with MySQL Enterprise Monitor User Interface:

- Microsoft Internet Explorer: version 11, and higher.
- Safari: most current major production release and one prior release
- Firefox: the most current major ESR version and above
- Google Chrome: the most current major stable channel release

C.3 Backing up MySQL Enterprise Service Manager

All of the MySQL Enterprise Monitor repository information is stored within a set of databases, one is simply named "mem", and the others are prefixed by "mem__", with a suffix denoting the type of data stored, such as "mem__config", "mem__instruments", "mem__events", and so on. There is also a MySQL schema which stores the privilege data which you must also back up.

The following is an example of how to use `mysqldump` to export all databases in the repository instance to a dump file named `mem.dump`:

```
Note
This example assumes you are using the MySQL instance solely as the repository for MySQL Enterprise Service Manager and not for any other purpose. This is the recommended implementation.
```

```
shell> mysqldump --hex-bin --single-transaction -uservice_manager -pPassword -P13306 -h127.0.0.1 —all-databases > mem.dump
```

The above command creates a file, `mem.dump`, containing all of the MySQL Enterprise Monitor data.

To restore the dump file, run the following `mysql` command on a clean instance:

```
shell> mysql -u <user> -p -P13306 -h127.0.0.1 < mem.dump
```

You should also backup the following files:

- `apache-tomcat/conf/ssl/tomcat.cert.pem`
- `apache-tomcat/conf/ssl/tomcat.key.pem`
- `apache-tomcat/conf/myKeystore`
- `java/lib/security/cacerts`
- `apache-tomcat/webapps/ROOT/WEB-INF/config.properties`
- `apache-tomcat/webapps/ROOT/WEB-INF/configArea/mem.keystore`

You can also use MySQL Enterprise Backup to manage your repository backup process. For more information, see the MySQL Enterprise Backup documentation.

To ensure consistency in a recovery situation, you may also want to back up the agent configuration and metadata stored on each monitored MySQL server. To do this:
• Copy the `etc` subdirectory of each agent to a backup location. This directory contains the configuration information for each server being monitored.

• On each server being monitored, back up the `mysql.inventory` table, which contains the unique ID of the MySQL server.

C.4 Regular MySQL Enterprise Monitor Maintenance

MySQL Enterprise Monitor is generally self-managing and does not need much maintenance. To keep your MySQL Enterprise Monitor running efficiently, automate or manually perform these maintenance tasks:

• Set the purge interval for your data to an appropriate value according to duration and history of data that you want to keep. For more information, see Data Purge Behavior.

• Check, and delete, the contents of the temporary directory with your MySQL Enterprise Service Manager installation directory.
Appendix D MySQL Enterprise Monitor Component Reference

Table of Contents

D.1 MySQL Enterprise Service Manager Reference ................................................................. 281
D.1.1 Log Files for the MySQL Enterprise Service Manager .................................................. 281
D.1.2 The Management Information Base (MIB) File .............................................................. 281
D.1.3 The config.properties file ................................................................................................. 282
D.1.4 Using The Service Monitor Repository Configuration Script ........................................... 286
D.2 MySQL Enterprise Monitor Agent Reference ......................................................................... 286
D.2.1 MySQL Enterprise Monitor Agent Configurable Options ............................................... 286
D.2.2 Agent Log Files ................................................................................................................. 286
D.2.3 Using The Agent Configuration Script ............................................................................... 287

D.1 MySQL Enterprise Service Manager Reference

D.1.1 Log Files for the MySQL Enterprise Service Manager

This section shows the location of the log files associated with the various components that make up the MySQL Enterprise Service Manager. These files can prove useful for debugging purposes.

All log files except catalina.out are rotated to ensure they do not grow beyond 10MB in size.

Table D.1 MySQL Enterprise Monitor: Log File Locations

<table>
<thead>
<tr>
<th>Component</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache/Tomcat</td>
<td>%MySQL\Enterprise\Monitor\apache-tomcat\logs\catalina.out</td>
</tr>
<tr>
<td>Repository</td>
<td>%MySQL\Enterprise\Monitor\mysql\data</td>
</tr>
<tr>
<td>Configuration Report</td>
<td>%MySQL\Enterprise\Monitor\configuration_report.txt</td>
</tr>
<tr>
<td>Service Manager (General)</td>
<td>%MySQL\Enterprise\Monitor\apache-tomcat\logs\mysql-monitor.log</td>
</tr>
<tr>
<td>Service Manager (Full / Support)</td>
<td>%MySQL\Enterprise\Monitor\apache-tomcat\logs\mysql-monitor-full.log</td>
</tr>
</tbody>
</table>

All of these logs are viewable within MySQL Enterprise Monitor User Interface. For more information, see Section 15.3, "Logs".

On all operating systems, the Apache/Tomcat, and Repository directories contain both access and error files.

D.1.2 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 D.2 MySQL Enterprise Monitor: MIB File Locations

<table>
<thead>
<tr>
<th>Operating System</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor\support-files\MONITOR.MIB</td>
</tr>
<tr>
<td>Unix</td>
<td>/opt/mysql/enterprise/monitor/support-files/MONITOR.MIB</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/monitor/support-files/MONITOR.MIB</td>
</tr>
</tbody>
</table>
D.1.3 The config.properties file

File location

The repository user name and encrypted password are stored in the config.properties file. The following table shows the location of this file on various operating systems:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\webapps\ROOT\WEB-INF</td>
</tr>
<tr>
<td>Linux and Unix</td>
<td>/opt/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF</td>
</tr>
</tbody>
</table>

Make sure that the file is secured at the filesystem level so that it cannot be read by anybody but the administrator, or MySQL Enterprise Monitor.

Usage

A generated config.properties file looks similar to:

```
#SymmetricKey was auto generated.
#Thu Aug 15 13:35:56 PDT 2013
mysql.use_ssl=true
mysql.user=service_manager
mysql.port=13306
key=8577667A79DF5275
mysql.pass=BMcsacZdrMmM7mrnFExURHDuxp4C3hcr2yxcpC2QpiE=
mysql.verify_server_cert=false
mysql.server=localhost
mysql.db=mem
```

Note

The mysql.pass is encrypted.

The application has two connection pools, one to service agent traffic, and the other for the UI. You can configure them as one logical pool with a 85/15 (agent/ui) percentage split, and use "dbPool" as the pool name in the further settings. Or, you can configure each pool separately, where the pool names are "default" and "ui". Note that the names after the "." come verbatim from DBCP at http://commons.apache.org/proper/commons-dbcp/configuration.html.

Table D.4 Optional config.properties values

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_collection_interval</td>
<td>string</td>
<td>00:01:00</td>
</tr>
<tr>
<td>dbpool.default.initialSize</td>
<td>integer</td>
<td>20</td>
</tr>
<tr>
<td>dbpool.default.maxActive</td>
<td>integer</td>
<td>70</td>
</tr>
<tr>
<td>dbpool.default.maxIdle</td>
<td>integer</td>
<td>20</td>
</tr>
</tbody>
</table>
The `config.properties` file

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dbpool.default.maxWaitMillis</code></td>
<td>string</td>
<td>30 seconds</td>
</tr>
<tr>
<td><code>dbpool.default.minIdle</code></td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td><code>dbpool.default.timeBetweenEvictionRunsMillis</code></td>
<td>string</td>
<td>5 seconds</td>
</tr>
<tr>
<td><code>dbpool.ui.initialSize</code></td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td><code>dbpool.ui.maxActive</code></td>
<td>integer</td>
<td>15</td>
</tr>
<tr>
<td><code>dbpool.ui.maxIdle</code></td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td><code>dbpool.ui.timeBetweenEvictionRunsMillis</code></td>
<td>string</td>
<td>5 seconds</td>
</tr>
<tr>
<td><code>internal_perf_enable</code></td>
<td>boolean</td>
<td>false</td>
</tr>
<tr>
<td><code>internal_perf_server_id</code></td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td><code>notify_thread_pool_size</code></td>
<td>integer</td>
<td>4</td>
</tr>
<tr>
<td><code>quanal.collect</code></td>
<td>string</td>
<td>00:01:00</td>
</tr>
<tr>
<td><code>supportReport.retention.minutes</code></td>
<td>string</td>
<td>6 hours</td>
</tr>
<tr>
<td><code>ui.javascript.useClientSideStorage</code></td>
<td>boolean</td>
<td>false</td>
</tr>
</tbody>
</table>

• `notify_thread_pool_size(4)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>integer</td>
<td></td>
</tr>
</tbody>
</table>

SMTP and SNMP notifications are sent asynchronously, this controls how many threads are used for this process.

• `thread_pool_size(8)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>integer</td>
<td></td>
</tr>
</tbody>
</table>

Used to handle background jobs.

• `data_collection_interval(00:01:00)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>string</td>
<td>00:01:00</td>
</tr>
</tbody>
</table>

Defaults to one minute, and is never less than one minute. May be set to a value larger than one minute by use of the `data_collection_interval` property, in `hh:mm:ss` interval format.

• `internal_perf_enable(false)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>boolean</td>
<td>false</td>
</tr>
</tbody>
</table>

Enables internal performance monitoring for MySQL Enterprise Monitor (requires deploying some graphs from the `resources/` directory).

• `internal_perf_server_id(false)`
The config.properties file

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If `internal_perf_enable` is set to true, and MySQL Enterprise Service Manager can not read `mysql.inventory`, then use this ID instead.

- `quanal.collect(00:01:00)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td></td>
<td>00:01:00</td>
</tr>
</tbody>
</table>

The rate that the service manager asks for query analysis data from the agent and plugins. It is expressed using the hh:mm:ss interval format.

- `ui.javascript.useClientSideStorage(false)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td></td>
<td>false</td>
</tr>
</tbody>
</table>

Use this instead of cookies to store UI state (not login, but graph selection, etc.) Generally only needed if using a broken proxy that truncates cookie length.

- `supportReport.retention.minutes(6 hours)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td></td>
<td>6 hours</td>
</tr>
</tbody>
</table>

The length of time that MySQL Enterprise Monitor will retain the reports generated when using "Support diagnostics" from "Manage Servers".

- `dbpool.ui.initialSize(5)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

- `dbpool.ui.maxActive(15)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer</td>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

- `dbpool.ui.minIdle(0)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

- `dbpool.ui.maxIdle(5)`

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

- `dbpool.ui.maxWaitMillis(30 seconds)`
The *config.properties* file

<table>
<thead>
<tr>
<th>Permitted Values</th>
<th>Type</th>
<th>Permitted Values</th>
<th>Type</th>
<th>Permitted Values</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpool.ui.timeBetweenEvictionRunsMillis(5 seconds)</td>
<td><strong>Default</strong> 30 seconds</td>
<td>dbpool.ui.minEvictableIdleTimeMillis(15 seconds)</td>
<td><strong>Default</strong> 5 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpool.default.initialSize(20)</td>
<td><strong>Default</strong> 20</td>
<td>dbpool.default.maxActive(70)</td>
<td><strong>Default</strong> 70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpool.default.minIdle(0)</td>
<td><strong>Default</strong> 0</td>
<td>dbpool.default.maxIdle(5)</td>
<td><strong>Default</strong> 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpool.default.maxWaitMillis(30 seconds)</td>
<td><strong>Default</strong> 30 seconds</td>
<td>dbpool.default.timeBetweenEvictionRunsMillis(5 seconds)</td>
<td><strong>Default</strong> 5 seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpool.default.minEvictableIdleTimeMillis(15 seconds)</td>
<td><strong>Default</strong> 15 seconds</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
D.1.4 Using The Service Monitor Repository Configuration Script

The config.sh/config.bat script is used to configure the MySQL Server Repository for the Service Monitor. Its default location:

Table D.5 MEM Repository Configuration Tool Location (default)

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor\bin\config.bat</td>
</tr>
<tr>
<td>Linux / Solaris</td>
<td>/opt/mysql/enterprise/monitor/bin/config.sh</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/monitor/bin/config.sh</td>
</tr>
</tbody>
</table>

Use --help to view its options:

```
bin/config.sh --help
Usage: java com.mysql.etools.agent.runtime.Main [options], where options are:

Option Description
----- -----------
--help Prints this usage message
--md, --mysql-db  MySQL database for the Service Manager repository
--mp, --mysql-port MySQL port for the Service Manager repository
--ms, --mysql-server MySQL server for the Service Manager repository
--mu, --mysql-user MySQL username for the Service Manager repository
-v, --version     Displays the version of the agent and components
```

An example use case is to move the MySQL Enterprise Service Manager Repository to its own host, as described under Tuning Performance: CPU.

D.2 MySQL Enterprise Monitor Agent Reference

The MySQL Enterprise Monitor Agent provides all of the information from a monitored MySQL server to MySQL Enterprise Service Manager.

The MySQL Enterprise Monitor Agent is configured through the MySQL Enterprise Monitor User Interface, and the bundled agent.sh/agent.bat script. Using these methods is recommended.

Note

In MEM versions before 3.0.0, the Agent was configured using the mysql-monitor-agent.ini and agent-instance.ini configuration files.

D.2.1 MySQL Enterprise Monitor Agent Configurable Options

The MySQL Enterprise Monitor Agent supports several configurable options that are configured using the bundled agent.sh or agent.bat script.

See agent.sh --help for details.

For usage information, see Section D.2.3, “Using The Agent Configuration Script”.

D.2.2 Agent Log Files

The Agent has two log files. mysql-monitor-agent.log is the general log, and mysql-monitor-agent-full.log is the full log that also contains stack traces that are useful to the Support team.
The default path to the Agent log files are as follows:

- **Windows Path:** `C:\Program Files\MySQL\Enterprise\Agent\logs\`
- **Linux Path:** `/opt/mysql/enterprise/agent/logs/`
- **Mac OS X Path:** `/Applications/mysql/enterprise/agent/logs/`

The log files are managed with log4j, which is configured using `log4j.properties`. The Agent watches for changes every 60 seconds, and updates MySQL Enterprise Monitor accordingly. The default file location:

- **Windows Path:** `C:\Program Files\MySQL\Enterprise\Agent\etc\log4j.properties`
- **Linux Path:** `/opt/mysql/enterprise/agent/etc/log4j.properties`
- **Mac OS X Path:** `/Applications/mysql/enterprise/agent/etc/log4j.properties`

The maximum size of a log file may be limited to 2GB. If MySQL Enterprise Monitor Agent cannot add information to the configured logfile, information is sent to the standard output instead.

Because the log files can become large, you could rotate the logs by defining log4j options. For example, to implement a rotation of 10 x 10MB log files:

```
log4j.appender.file.MaximumFileSize = 10MB
log4j.appender.file.MaxBackupIndex = 10
log4j.appender.file.Append = true
```

For additional information about log4j, read the log4j documentation at [http://logging.apache.org/log4j/](http://logging.apache.org/log4j/).

### D.2.3 Using The Agent Configuration Script

The `agent.sh / agent.bat` script is used to configure an Agent. Its location:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td><code>C:\Program Files\MySQL\Enterprise\Agent\bin\agent.bat</code></td>
</tr>
<tr>
<td>Linux / Solaris</td>
<td><code>/opt/mysql/enterprise/agent/bin/agent.sh</code></td>
</tr>
<tr>
<td>Mac OS X</td>
<td><code>/Applications/mysql/enterprise/agent/bin/agent.sh</code></td>
</tr>
</tbody>
</table>

Use `--help` to view its options.

This script is used to handle several use cases, including:

- **Starting the Agent daemon/service:** Execute the script without any parameters.
- **Configuring bootstrap.properties:** Execute the script using the `--url` or `--agent-user` options.
- **Managing a connection (add/edit/delete):** Execute the script using the `--create-connection`, `--edit-connection`, or `--remove-connection` options.
- **Assigning an Agent to multiple groups:** Execute the script using multiple iterations of the `--connection-group` option, one for each assigned group.

**Note**

Executing a use case is exclusive, as each use case requires its own execution of the script.
Appendix E Error codes

The following table documents error codes reported by the GUI and the messages associated with each code.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>U0001</td>
<td>&quot;replaceable_value&quot; is a required field.</td>
</tr>
<tr>
<td>U0002</td>
<td>You must log in to access the requested resource.</td>
</tr>
<tr>
<td>U0003</td>
<td>You do not have permissions to access this resource.</td>
</tr>
<tr>
<td>U0004</td>
<td>The user field &quot;replaceable_value&quot; must not be null. Either the submission was broken or you have accessed this resource improperly.</td>
</tr>
<tr>
<td>U0005</td>
<td>A non-numeric value was submitted for the field &quot;replaceable_value&quot;. Either the submission was broken or you have accessed this resource improperly.</td>
</tr>
<tr>
<td>U0007</td>
<td>Unable to connect to download Advisors. replaceable_value</td>
</tr>
<tr>
<td>U0009</td>
<td>The uploaded Advisor jar was invalid.</td>
</tr>
<tr>
<td>U0010</td>
<td>The username and password combination was not found.</td>
</tr>
<tr>
<td>U0011</td>
<td>The user &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>U0012</td>
<td>You may not log into the interface with agent credentials.</td>
</tr>
<tr>
<td>U0013</td>
<td>At least one user with the &quot;agent&quot; role is required to use the application.</td>
</tr>
<tr>
<td>U0014</td>
<td>At least one user with the &quot;manager&quot; role is required to use the application.</td>
</tr>
<tr>
<td>U0018</td>
<td>You must provide an admin password.</td>
</tr>
<tr>
<td>U0019</td>
<td>The admin passwords do not match.</td>
</tr>
<tr>
<td>U0020</td>
<td>You must provide an agent password.</td>
</tr>
<tr>
<td>U0021</td>
<td>The agent passwords do not match.</td>
</tr>
<tr>
<td>U0023</td>
<td>You must provide a non-zero interval.</td>
</tr>
<tr>
<td>U0024</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;To&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0025</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;From&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0026</td>
<td>Your &quot;To&quot; and &quot;From&quot; dates cannot be the same date.</td>
</tr>
<tr>
<td>U0027</td>
<td>You must provide a graph id for display.</td>
</tr>
<tr>
<td>U0028</td>
<td>You have exceeded the maximum number of graphs for display. You may only select replaceable_value graphs.</td>
</tr>
<tr>
<td>U0029</td>
<td>No graphs ids were found.</td>
</tr>
<tr>
<td>U0030</td>
<td>You must provide a non-zero interval.</td>
</tr>
<tr>
<td>U0031</td>
<td>Your full-sized width must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0032</td>
<td>&quot;replaceable_value&quot; is not a valid value for full-sized width.</td>
</tr>
<tr>
<td>U0033</td>
<td>Your full-sized height must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0034</td>
<td>&quot;replaceable_value&quot; is not a valid value for full-sized height.</td>
</tr>
<tr>
<td>U0035</td>
<td>Your thumbnail width must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0036</td>
<td>&quot;replaceable_value&quot; is not a valid value for thumbnail width.</td>
</tr>
<tr>
<td>U0037</td>
<td>Your thumbnail height must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0038</td>
<td>&quot;replaceable_value&quot; is not a valid value for thumbnail height.</td>
</tr>
<tr>
<td>U0039</td>
<td>You must select one or more events to close.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>U0041</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;From&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0042</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;To&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0043</td>
<td>You must provide a non-zero interval.</td>
</tr>
<tr>
<td>U0044</td>
<td>Your width must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0045</td>
<td>&quot;replaceable_value&quot; is not a valid value for width.</td>
</tr>
<tr>
<td>U0046</td>
<td>Your height must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0047</td>
<td>&quot;replaceable_value&quot; is not a valid value for height.</td>
</tr>
<tr>
<td>U0048</td>
<td>You must provide a non-zero interval.</td>
</tr>
<tr>
<td>U0049</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;To&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0050</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;From&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0051</td>
<td>Your &quot;To&quot; and &quot;From&quot; dates cannot be the same date.</td>
</tr>
<tr>
<td>U0052</td>
<td>Your width must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0053</td>
<td>&quot;replaceable_value&quot; is not a valid value for width.</td>
</tr>
<tr>
<td>U0054</td>
<td>Your height must be greater than or equal to replaceable_value.</td>
</tr>
<tr>
<td>U0055</td>
<td>&quot;replaceable_value&quot; is not a valid value for height.</td>
</tr>
<tr>
<td>U0056</td>
<td>You must select one or more advisors.</td>
</tr>
<tr>
<td>U0057</td>
<td>No schedule was found with id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0058</td>
<td>You must select one or more rules to schedule.</td>
</tr>
<tr>
<td>U0059</td>
<td>There was a problem scheduling &quot;replaceable_value&quot; against &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0060</td>
<td>No rule was found with the uuid &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0061</td>
<td>Advisor name must not be empty.</td>
</tr>
<tr>
<td>U0062</td>
<td>Advisor &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>U0063</td>
<td>New advisor name must not be empty.</td>
</tr>
<tr>
<td>U0064</td>
<td>Advisor id must not be null.</td>
</tr>
<tr>
<td>U0065</td>
<td>No advisor was found with the id replaceable_value.</td>
</tr>
<tr>
<td>U0066</td>
<td>Rule uuid must not be null.</td>
</tr>
<tr>
<td>U0067</td>
<td>No rule was found with the uuid &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0068</td>
<td>One or more variables is required to make a functioning rule.</td>
</tr>
<tr>
<td>U0069</td>
<td>Each defined variable must be mapped to an instance. The variable &quot;replaceable_value&quot; is missing the instance to collect against.</td>
</tr>
<tr>
<td>U0070</td>
<td>Please choose an existing Advisor or create a new one with a non-empty name.</td>
</tr>
<tr>
<td>U0071</td>
<td>The default frequency must be a non-zero value.</td>
</tr>
<tr>
<td>U0072</td>
<td>You must enter a rule name.</td>
</tr>
<tr>
<td>U0075</td>
<td>You must specify a file for upload.</td>
</tr>
<tr>
<td>U0077</td>
<td>You must provide an email address.</td>
</tr>
<tr>
<td>U0078</td>
<td>The provided email address appears to be invalid.</td>
</tr>
<tr>
<td>U0079</td>
<td>A valid SMTP server address must be given.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>U0080</td>
<td>Your passwords do not match.</td>
</tr>
<tr>
<td>U0081</td>
<td>A valid SNMP target must be given.</td>
</tr>
<tr>
<td>U0082</td>
<td>A valid SNMP port must be given.</td>
</tr>
<tr>
<td>U0083</td>
<td>A valid SNMP community string must be given.</td>
</tr>
<tr>
<td>U0084</td>
<td>The given file could not be found.</td>
</tr>
<tr>
<td>U0086</td>
<td>An invalid file name was given.</td>
</tr>
<tr>
<td>U0088</td>
<td>An invalid SNMP severity level was given.</td>
</tr>
<tr>
<td>U0090</td>
<td>An invalid data life span was given.</td>
</tr>
<tr>
<td>U0091</td>
<td>An invalid re-inventory frequency was given.</td>
</tr>
<tr>
<td>U0093</td>
<td>The user name is missing.</td>
</tr>
<tr>
<td>U0095</td>
<td>An invalid time zone was given.</td>
</tr>
<tr>
<td>U0096</td>
<td>An invalid locale was given.</td>
</tr>
<tr>
<td>U0097</td>
<td>The group name is invalid.</td>
</tr>
<tr>
<td>U0098</td>
<td>The recipients list is empty.</td>
</tr>
<tr>
<td>U0099</td>
<td>Cannot find the e-mail target.</td>
</tr>
<tr>
<td>U0100</td>
<td>The given target id is invalid.</td>
</tr>
<tr>
<td>U0101</td>
<td>The group name must not be empty.</td>
</tr>
<tr>
<td>U0102</td>
<td>The server name must not be empty.</td>
</tr>
<tr>
<td>U0103</td>
<td>An invalid group id was given.</td>
</tr>
<tr>
<td>U0104</td>
<td>The group already exists.</td>
</tr>
<tr>
<td>U0105</td>
<td>This group name is already in use. Enter a different name.</td>
</tr>
<tr>
<td>U0106</td>
<td>Cannot find group with id replaceable_value.</td>
</tr>
<tr>
<td>U0107</td>
<td>Cannot find server with id replaceable_value.</td>
</tr>
<tr>
<td>U0108</td>
<td>An invalid server id was given.</td>
</tr>
<tr>
<td>U0109</td>
<td>Invalid Username</td>
</tr>
<tr>
<td>U0110</td>
<td>Invalid user role specified.</td>
</tr>
<tr>
<td>U0111</td>
<td>A user with the given name already exists.</td>
</tr>
<tr>
<td>U0112</td>
<td>The password must not be empty.</td>
</tr>
<tr>
<td>U0113</td>
<td>The user &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>U0114</td>
<td>The log name must not be empty.</td>
</tr>
<tr>
<td>U0115</td>
<td>An invalid log level was given.</td>
</tr>
<tr>
<td>U0117</td>
<td>An invalid filter was set.</td>
</tr>
<tr>
<td>U0118</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;From&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0119</td>
<td>&quot;replaceable_value&quot; is an invalid &quot;To&quot; date. Dates are in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>U0120</td>
<td>No event was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0121</td>
<td>No category was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0122</td>
<td>You must select one or more instances to add.</td>
</tr>
<tr>
<td>U0123</td>
<td>No log was found with the name &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0124</td>
<td>No schedule was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>U0125</td>
<td>No notification group was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0126</td>
<td>Unable to retrieve Advisors at this time.</td>
</tr>
<tr>
<td>U0128</td>
<td>You must specify a file for upload.</td>
</tr>
<tr>
<td>U0129</td>
<td>The uploaded Graph XML was invalid.</td>
</tr>
<tr>
<td>U0130</td>
<td>The uploaded Graph had an insufficient version number, and was not loaded.</td>
</tr>
<tr>
<td>U0131</td>
<td>No group was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0132</td>
<td>No user was found with the name &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0133</td>
<td>No log was found with the key &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0134</td>
<td>An invalid e-mail address was given.</td>
</tr>
<tr>
<td>U0135</td>
<td>No user was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0136</td>
<td>No graph was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0137</td>
<td>You must provide a query summary id for display.</td>
</tr>
<tr>
<td>U0138</td>
<td>No query summary was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0139</td>
<td>&quot;replaceable_value&quot; is not a valid UUID.</td>
</tr>
<tr>
<td>U0140</td>
<td>You are not authorized to access this resource.</td>
</tr>
<tr>
<td>U0141</td>
<td>No graph was found with the uuid &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0142</td>
<td>The target name is already in use.</td>
</tr>
<tr>
<td>U0143</td>
<td>No key was provided to sort on.</td>
</tr>
<tr>
<td>U0145</td>
<td>U0120 Unable to import Advisor Jar. You may download the jar manually from the My Oracle Support website and import it on this page.</td>
</tr>
<tr>
<td>U0146</td>
<td>Unable to schedule rule &quot;replaceable_value&quot; due to &quot;replaceable_value.replaceable_value&quot; data not being collected from server &quot;replaceable_value&quot;. It may be an unsupported collection for that server.</td>
</tr>
<tr>
<td>U0147</td>
<td>The requested feature, replaceable_value, has not been implemented.</td>
</tr>
<tr>
<td>U0148</td>
<td>You must select one or more servers.</td>
</tr>
<tr>
<td>U0149</td>
<td>You must provide a non-empty alias.</td>
</tr>
<tr>
<td>U0150</td>
<td>No instance was found with the id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0151</td>
<td>Instance already exists with name &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0152</td>
<td>No instance name provided</td>
</tr>
<tr>
<td>U0154</td>
<td>Query Analyzer Data purge value cannot be greater than Historic Data purge value.</td>
</tr>
<tr>
<td>U0156</td>
<td>The database cannot parse the regular expression: replaceable_value.</td>
</tr>
<tr>
<td>U0157</td>
<td>Query Explain support requires Query Example capture to be enabled.</td>
</tr>
<tr>
<td>U0162</td>
<td>The admin and agent usernames must be different.</td>
</tr>
<tr>
<td>U0163</td>
<td>The requested alias is too long; aliases must be less than replaceable_value characters.</td>
</tr>
<tr>
<td>U0164</td>
<td>&quot;replaceable_value&quot; is not a DNS resolvable hostname or IP address.</td>
</tr>
<tr>
<td>U0165</td>
<td>Heat chart rule &quot;replaceable_value&quot; scheduled against &quot;replaceable_value&quot; may not be unscheduled. This rule is required for the product to function properly.</td>
</tr>
<tr>
<td>U0166</td>
<td>Heat chart rule &quot;replaceable_value&quot; scheduled against &quot;replaceable_value&quot; may not be disabled. This rule is required for the product to function properly.</td>
</tr>
<tr>
<td>U0167</td>
<td>Bad graph identifier.</td>
</tr>
<tr>
<td>U0168</td>
<td>There is no graph with the given identifier.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>U0169</td>
<td>Graph width &quot;replaceable_value&quot; is invalid.</td>
</tr>
<tr>
<td>U0170</td>
<td>Graph height &quot;replaceable_value&quot; is invalid.</td>
</tr>
<tr>
<td>U0171</td>
<td>Graph width &quot;replaceable_value&quot; is below minimum &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0172</td>
<td>Graph height &quot;replaceable_value&quot; is below minimum &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0173</td>
<td>Interval &quot;replaceable_value&quot; is not valid.</td>
</tr>
<tr>
<td>U0175</td>
<td>A HTTP proxy host is required; the port is optional.</td>
</tr>
<tr>
<td>U0176</td>
<td>Unable to parse the HTTP host &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0177</td>
<td>Unable to parse the HTTP proxy port number &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>U0178</td>
<td>An invalid hostname or port was given.</td>
</tr>
<tr>
<td>U0179</td>
<td>The HTTP proxy passwords must match.</td>
</tr>
<tr>
<td>U0180</td>
<td>You cannot change the role of the logged-in user.</td>
</tr>
<tr>
<td>U0181</td>
<td>No updates are available to install at this time.</td>
</tr>
<tr>
<td>U0182</td>
<td>There was an error formatting the Advisor Readme.</td>
</tr>
<tr>
<td>U0183</td>
<td>You must provide a primary hostname.</td>
</tr>
<tr>
<td>U0186</td>
<td>You must provide a login when not using anonymous binds.</td>
</tr>
<tr>
<td>U0187</td>
<td>You must provide a password attribute when using comparison mode.</td>
</tr>
<tr>
<td>U0188</td>
<td>You must provide a password digest when using comparison mode.</td>
</tr>
<tr>
<td>U0189</td>
<td>Unknown password digest provided.</td>
</tr>
<tr>
<td>U0191</td>
<td>User or role pattern required when using role mapping.</td>
</tr>
<tr>
<td>U0192</td>
<td>You must provide a role search pattern attribute.</td>
</tr>
<tr>
<td>U0193</td>
<td>You must provide at least one mapping for the &quot;admin&quot; role.</td>
</tr>
<tr>
<td>U0194</td>
<td>You must provide at least one mapping for the &quot;dba&quot; role.</td>
</tr>
<tr>
<td>U0195</td>
<td>You must provide at least one mapping for the &quot;agent&quot; role.</td>
</tr>
<tr>
<td>U0201</td>
<td>You must provide at least one mapping for the &quot;read-only&quot; role.</td>
</tr>
<tr>
<td>U0202</td>
<td>replaceable_value heat chart {0,choice,1#rule</td>
</tr>
<tr>
<td>U0203</td>
<td>No schedules were found for the rule.</td>
</tr>
<tr>
<td>U0204</td>
<td>Delete without confirmation disallowed.</td>
</tr>
<tr>
<td>U0205</td>
<td>Nothing is selected.</td>
</tr>
<tr>
<td>U0206</td>
<td>No file was provided.</td>
</tr>
<tr>
<td>U0207</td>
<td>Unable to determine the type of this file.</td>
</tr>
<tr>
<td>U0208</td>
<td>Nothing is selected.</td>
</tr>
<tr>
<td>U0209</td>
<td>The selection is invalid.</td>
</tr>
<tr>
<td>U0210</td>
<td>A version (e.g., &quot;1.0&quot;) is required.</td>
</tr>
<tr>
<td>U0211</td>
<td>replaceable_value heat chart {0,choice,1#rule</td>
</tr>
<tr>
<td>U0212</td>
<td>You must specify user search criteria.</td>
</tr>
<tr>
<td>U0213</td>
<td>You must choose user search by pattern or by criteria.</td>
</tr>
</tbody>
</table>
The following table documents error codes reported by the Enterprise server and the messages associated with each code.

**Table E.2 MySQL Enterprise Monitor: Server Codes and Messages**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>E0001</td>
<td>Internal Error: <code>replaceable_value</code></td>
</tr>
<tr>
<td>E0100</td>
<td>Invalid user role &quot;<code>replaceable_value</code>&quot;. Only &quot;manager&quot;, &quot;dba&quot;, and &quot;agent&quot; are valid.</td>
</tr>
<tr>
<td>E0101</td>
<td>Invalid user name &quot;<code>replaceable_value</code>&quot;. User names must be alphanumeric and between 1-32 characters in length.</td>
</tr>
<tr>
<td>E0102</td>
<td>Invalid user password. Passwords must contain at least one number.</td>
</tr>
<tr>
<td>E0103</td>
<td>Cannot delete manager user.</td>
</tr>
<tr>
<td>E0104</td>
<td>User cannot change role.</td>
</tr>
<tr>
<td>E0105</td>
<td>User &quot;<code>replaceable_value</code>&quot; already exists.</td>
</tr>
<tr>
<td>E0106</td>
<td>User &quot;<code>replaceable_value</code>&quot; does not exist.</td>
</tr>
<tr>
<td>E0107</td>
<td>User role &quot;<code>replaceable_value</code>&quot; is not authorized.</td>
</tr>
<tr>
<td>E0108</td>
<td>User &quot;<code>replaceable_value</code>&quot; has been previously deleted and is disabled.</td>
</tr>
<tr>
<td>E0200</td>
<td>No mapping exists to map the user &quot;<code>replaceable_value</code>&quot; to a MySQL Network user.</td>
</tr>
<tr>
<td>E0201</td>
<td>A mapping already exists for the user &quot;<code>replaceable_value</code>&quot; to a MySQL Network user.</td>
</tr>
<tr>
<td>E0202</td>
<td>MySQL Network is currently unavailable or could not be reached.</td>
</tr>
<tr>
<td>E0203</td>
<td>MySQL Network reported no alerts.</td>
</tr>
<tr>
<td>E0204</td>
<td>MySQL Network Advisor returned no data. Reason: &quot;<code>replaceable_value</code>&quot;.</td>
</tr>
<tr>
<td>E0205</td>
<td>Your Advisors are currently up to date.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>E0206</td>
<td>Your MySQL Enterprise trial expired after &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0207</td>
<td>The MySQL Enterprise subscription key is not valid.</td>
</tr>
<tr>
<td>E0208</td>
<td>No MySQL Enterprise subscription information is installed.</td>
</tr>
<tr>
<td>E0209</td>
<td>The MySQL Enterprise subscription key has expired.</td>
</tr>
<tr>
<td>E0210</td>
<td>User: replaceable_value Requires role: replaceable_value</td>
</tr>
<tr>
<td>E0211</td>
<td>You do not have permissions to access this resource.</td>
</tr>
<tr>
<td>E0212</td>
<td>The import of a duplicate MySQL Enterprise subscription trial key was rejected.</td>
</tr>
<tr>
<td>E0213</td>
<td>Unable to decode key from file: replaceable_value.</td>
</tr>
<tr>
<td>E0300</td>
<td>Can not delete default group.</td>
</tr>
<tr>
<td>E0301</td>
<td>Group &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E0302</td>
<td>Group &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0303</td>
<td>Invalid group name &quot;replaceable_value&quot;. Group name must be alphanumeric and less than 128 characters.</td>
</tr>
<tr>
<td>E0304</td>
<td>Group element &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E0305</td>
<td>Group element &quot;replaceable_value&quot; does not exist in the group &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0400</td>
<td>Agent &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0401</td>
<td>Agent payload parameter NULL.</td>
</tr>
<tr>
<td>E0402</td>
<td>Agent task &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0403</td>
<td>There has been a problem de-serializing this agent&quot;s request. Please ensure all agents are version replaceable_value or greater.</td>
</tr>
<tr>
<td>E0500</td>
<td>Server &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E0501</td>
<td>Server &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0502</td>
<td>Server &quot;replaceable_value&quot; is actively being monitored and cannot be deleted. The last agent contact for this server was on replaceable_value.</td>
</tr>
<tr>
<td>E0503</td>
<td>Server &quot;replaceable_value&quot; is being migrated and cannot be deleted. Please halt the active migration first.</td>
</tr>
<tr>
<td>E0600</td>
<td>Database &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E0601</td>
<td>Database &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0700</td>
<td>Threshold &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E0701</td>
<td>Threshold &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0702</td>
<td>Data collection item does not exist.</td>
</tr>
<tr>
<td>E0703</td>
<td>serverName or groupName have to be set.</td>
</tr>
<tr>
<td>E0800</td>
<td>Data collection item does not exist.</td>
</tr>
<tr>
<td>E0801</td>
<td>Data collection is already occuring for &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0802</td>
<td>Data collection does not exist for &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0803</td>
<td>Invalid lifespan. Data life must be greater than 0 days.</td>
</tr>
<tr>
<td>E0804</td>
<td>Invalid namespace. Supported namespaces include &quot;mysql&quot; and &quot;os&quot;. replaceable_value</td>
</tr>
<tr>
<td>E0805</td>
<td>Invalid instance. The instance must be a valid for data collection.</td>
</tr>
<tr>
<td>E0806</td>
<td>Invalid source. The source must be valid for data collection.</td>
</tr>
<tr>
<td>E0807</td>
<td>Invalid attribute. The attrib must be valid for data collection.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>E0808</td>
<td>Invalid frequency: &quot;replaceable_value&quot;. The frequency must be at least 1 minute and be in the format HH:MM.</td>
</tr>
<tr>
<td>E0809</td>
<td>Invalid port. If the URI uses the MYSQL namespace it must include a server port.</td>
</tr>
<tr>
<td>E0810</td>
<td>Invalid server. The server &quot;replaceable_value&quot; must exist in the current inventory.</td>
</tr>
<tr>
<td>E0811</td>
<td>Invalid server. For an OS namespace, the server must not include the MySQL server port.</td>
</tr>
<tr>
<td>E0812</td>
<td>A SQL failure occurred while saving the data collection schedule. &quot;replaceable_value&quot;</td>
</tr>
<tr>
<td>E0813</td>
<td>Please see the error logs for details.</td>
</tr>
<tr>
<td>E0814</td>
<td>Invalid server id &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0815</td>
<td>No data items exist for server &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0816</td>
<td>Value &quot;replaceable_value&quot; not valid for &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0817</td>
<td>categoryName has to be set</td>
</tr>
<tr>
<td>E0818</td>
<td>The server &quot;replaceable_value&quot; does not support item(s) &quot;replaceable_value&quot; required by monitor &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0819</td>
<td>Invalid data type: &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0900</td>
<td>Advisor id &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0901</td>
<td>Advisor name &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0902</td>
<td>Advisor XML is invalid.</td>
</tr>
<tr>
<td>E0903</td>
<td>Category &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E0904</td>
<td>Category &quot;replaceable_value&quot; is incompatible with server version &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E0905</td>
<td>SKIPPED &quot;replaceable_value&quot; for replaceable_value because Agent for replaceable_value is down.</td>
</tr>
<tr>
<td>E0906</td>
<td>SKIPPED &quot;replaceable_value&quot; for replaceable_value because mysqld on replaceable_value is down.</td>
</tr>
<tr>
<td>E0907</td>
<td>Category &quot;replaceable_value&quot; cannot be modified.</td>
</tr>
<tr>
<td>E0908</td>
<td>Category &quot;replaceable_value&quot; cannot be deleted.</td>
</tr>
<tr>
<td>E0909</td>
<td>Category &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E0910</td>
<td>Advisor bundle version replaceable_value is incompatible with server version replaceable_value.</td>
</tr>
<tr>
<td>E0911</td>
<td>Heat chart rule &quot;replaceable_value&quot; scheduled against &quot;replaceable_value&quot; may not be unscheduled.</td>
</tr>
<tr>
<td>E0912</td>
<td>Heat chart rule &quot;replaceable_value&quot; scheduled against &quot;replaceable_value&quot; may not be disabled.</td>
</tr>
<tr>
<td>E1100</td>
<td>Graph time format &quot;replaceable_value&quot; is invalid. Must be in the format HH:MM:SS.</td>
</tr>
<tr>
<td>E1101</td>
<td>Graph size height and width must be specified.</td>
</tr>
<tr>
<td>E1102</td>
<td>Graph &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E1103</td>
<td>Graph &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E1104</td>
<td>Graph &quot;replaceable_value&quot; contains a data collection item that can not be identified.</td>
</tr>
<tr>
<td>E1200</td>
<td>Failed to retrieve identity column.</td>
</tr>
<tr>
<td>E1201</td>
<td>Required parameter &quot;replaceable_value&quot; was NULL.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>E1202</td>
<td>Only SELECT and SHOW commands are allowed via this interface.</td>
</tr>
<tr>
<td>E1203</td>
<td>Invalid timezone &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1204</td>
<td>Invalid interval &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1205</td>
<td>Could not find object &quot;replaceable_value&quot; in cache &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1206</td>
<td>Feature is not implemented. Parameters: replaceable_value</td>
</tr>
<tr>
<td>E1300</td>
<td>Advisor &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E1301</td>
<td>Advisor &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E1302</td>
<td>Schedule does not exist. &quot;replaceable_value&quot; &quot;replaceable_value&quot;</td>
</tr>
<tr>
<td>E1303</td>
<td>JEP error: expression : &quot;replaceable_value&quot;, message: &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1304</td>
<td>Advisor &quot;replaceable_value&quot; contains a data collection item that can not be identified.</td>
</tr>
<tr>
<td>E1305</td>
<td>Alarm level &quot;replaceable_value&quot; is not valid.</td>
</tr>
<tr>
<td>E1306</td>
<td>Schedule already exists. &quot;replaceable_value&quot; &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1307</td>
<td>Advisor &quot;replaceable_value&quot; is a read only MySQL Network certified Advisor. Please copy the rule before editing.</td>
</tr>
<tr>
<td>E1308</td>
<td>Advisor &quot;replaceable_value&quot; is currently scheduled against one or more monitored MySQL servers and cannot be removed.</td>
</tr>
<tr>
<td>E1309</td>
<td>Could not render text. &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1310</td>
<td>No open event for &quot;replaceable_value&quot; on server &quot;replaceable_value&quot;. Perhaps already closed.</td>
</tr>
<tr>
<td>E1311</td>
<td>Alarm with id &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E1400</td>
<td>Invalid missed agent heartbeat threshold value.</td>
</tr>
<tr>
<td>E1401</td>
<td>Notification entry &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E1402</td>
<td>Agent &quot;replaceable_value&quot; is using session &quot;replaceable_value&quot; but session &quot;replaceable_value&quot; was requested.</td>
</tr>
<tr>
<td>E1403</td>
<td>Server name &quot;replaceable_value&quot; is in use by another agent with uuid &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1500</td>
<td>Notification target &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E1501</td>
<td>Invalid email address &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1502</td>
<td>Notification target &quot;replaceable_value&quot; already exists.</td>
</tr>
<tr>
<td>E1503</td>
<td>Invalid notification target name &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1504</td>
<td>Invalid monitor name, it must not be empty.</td>
</tr>
<tr>
<td>E1505</td>
<td>No variable assignments given, you must define at least one.</td>
</tr>
<tr>
<td>E1506</td>
<td>SMTP authentication failed.</td>
</tr>
<tr>
<td>E1507</td>
<td>SMTP send failed</td>
</tr>
<tr>
<td>E1508</td>
<td>Invalid SNMP target &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1509</td>
<td>Invalid SNMP trap type &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1600</td>
<td>Log &quot;replaceable_value&quot; does not exist.</td>
</tr>
<tr>
<td>E1700</td>
<td>Authentication Mechanism was null</td>
</tr>
<tr>
<td>E1701</td>
<td>Unsupported Authentication Mechanism replaceable_value</td>
</tr>
<tr>
<td>E1702</td>
<td>Incorrect password for user replaceable_value</td>
</tr>
<tr>
<td>E1800</td>
<td>Invalid data type: &quot;replaceable_value&quot;, new value: &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>E1801</td>
<td>Invalid value: &quot;replaceable_value&quot; for data type &quot;replaceable_value&quot; for item &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1802</td>
<td>Invalid expression: &quot;replaceable_value&quot;, exception: &quot;replaceable_value&quot;, raw expression: &quot;replaceable_value&quot;. Substitute values: &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E1900</td>
<td>History not found id: &quot;replaceable_value&quot;, schedule: &quot;replaceable_value&quot;, expression: &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E2000</td>
<td>Tag not found, id: &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E2101</td>
<td>Unable to collect &quot;replaceable_value&quot; on &quot;replaceable_value&quot; for instance &quot;replaceable_value&quot;.</td>
</tr>
<tr>
<td>E9000</td>
<td>replaceable_value</td>
</tr>
<tr>
<td>E9001</td>
<td>replaceable_value from: replaceable_value</td>
</tr>
</tbody>
</table>
Appendix F Files associated with configuring, debugging, and using MySQL Enterprise Monitor

This links to the sections that discuss MySQL Enterprise Monitor generated files.

- Log files for the MySQL Enterprise Service Manager which includes the `mysql-monitor.log` and `mysql-monitor-full.log` logs (Service Manager), `catalina.out` (Apache/Tomcat), and `configuration_report.txt` (during the installation or upgrade). See Section D.1.1, “Log Files for the MySQL Enterprise Service Manager”.

- Agent log and pid files, which includes `mysql-monitor-agent.log` and `mysql-monitor-agent.pid`. See Section D.2.2, “Agent Log Files”.

- The Management Information Base (MIB) file, which is `MONITOR.MIB`. See Section D.1.2, “The Management Information Base (MIB) File”.

- The Service Manager configuration file, which is `config.properties`. See Section D.1.3, “The config.properties file”.

- The Service Manager may be started/stopped with `mysqlmonitorctl.sh` on Linux and Mac OS X, and `mysqlmonitorctl.bat` on Microsoft Windows.

![Important]

The `mysqlmonitorctl` file uses the locally-bound port 18005 to start/stop Tomcat.

- The agent may be started/stopped with `mysql-monitor-agent` on Mac OS X, `mysql-monitor-agent` on Linux, and `agentctl.bat` on Microsoft Windows.
Appendix G Data Collection Items

Note
MySQL Enterprise Monitor is available as part of the MySQL Enterprise
subscription, learn more at http://www.mysql.com/products/.

This appendix documents the data collection items used to create rules and graphs. 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 22.2, “Creating Advisors and Rules”.

Table G.1 MySQL Monitor Data Items

<table>
<thead>
<tr>
<th>Namespace</th>
<th>Namespace Type</th>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_dataitem_spec_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_dataitem_spec_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_attribute_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_attribute_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_class_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_class_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_class_config_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_class_config_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_instance_name_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_instance_name_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_classes_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_classes_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_instance_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_instance_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_namespaces_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_namespaces_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_value_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_item_value_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_mysqld_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_mysqld_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_plugin_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_plugin_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_target_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_target_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_task_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>agent_task_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>curl_mem_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>curl_mem_bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>curl_mem_bytes_max</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>curl_mem_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_data_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
<td>----------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_data_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_datum_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_datum_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_string_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_response_string_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_task_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>job_task_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>network_backlog_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>network_backlog_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>network_backlog_node_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>network_backlog_node_bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>network_backlog_node_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>scheduler_job_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>scheduler_job_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>xml_mem_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>xml_mem_bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>xml_mem_bytes_max</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>allocationstats</td>
<td>xml_mem_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>glib_mem_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>glib_mem_bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>glib_mem_bytes_max</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>glib_mem_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>lua_mem_alloc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>lua_mem_bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>lua_mem_bytes_max</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>chassisstats</td>
<td>lua_mem_free</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>lua</td>
<td>mem_size</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>egid</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>euid</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>gid</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>mem_major_faults</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>mem_minor_faults</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>mem_page_faults</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>mem_resident</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>mem_share</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>mem_size</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>nice</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>ppid</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>agent</td>
<td>proc</td>
<td>priority</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>processor</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>state</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>threads</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>time_start_time</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>time_sys</td>
<td>long_counter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>time_total</td>
<td>long_counter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>time_user</td>
<td>long_counter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>tty</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agent proc</td>
<td>uid</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>monitor resourceBundle</td>
<td>subscription</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>monitor resourceBundle</td>
<td>version</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql account_old_password</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql active_count</td>
<td>active_count</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql Agent</td>
<td>agent.latency</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql Agent</td>
<td>agentreachable</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql Agent</td>
<td>host_id</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql Agent</td>
<td>name</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql Agent</td>
<td>quanDataOutOfBand</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql Agent</td>
<td>version</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql anonymous_user</td>
<td>user_count</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql auto_increment_limit</td>
<td>table_list</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql avail_count</td>
<td>avail_count</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql broad_host_specifier</td>
<td>user</td>
<td>string</td>
<td>A list of users whose host in the mysql.user table meets the condition: WHERE host = '%'; a single string in wiki markup format.</td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_data_memory</td>
<td>data_mem_total</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_data_memory</td>
<td>data_mem_used</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_index_memory</td>
<td>index_mem_total</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_index_memory</td>
<td>index_mem_used</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_redo_buffer</td>
<td>redo_buffer_total</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_redo_buffer</td>
<td>redo_buffer_used</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_undo_buffer</td>
<td>undo_buffer_total</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_undo_buffer</td>
<td>undo_buffer_used</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_undo_logspace</td>
<td>undo_logspace_total</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql cluster_data_node_undo_logspace</td>
<td>undo_logspace_used</td>
<td>long</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

303
<table>
<thead>
<tr>
<th>Namespace Type</th>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql</td>
<td>cluster_data_node_undo_logspace</td>
<td>undo_logspace_used</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>cluster_data_nodes</td>
<td>not_started</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>column</td>
<td>Default</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>column</td>
<td>Extra</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>column</td>
<td>Field</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>column</td>
<td>Key</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>column</td>
<td>Null</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>column</td>
<td>Type</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>committed_count</td>
<td>committed_count</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>Database</td>
<td>Database</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Database</td>
<td>name</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>extra</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>id</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>key</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>key_len</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>possible_keys</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>ref</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>rows</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>select_type</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>table</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>Explain</td>
<td>type</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_database_io</td>
<td>logical_reads</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_database_io</td>
<td>physical_reads</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_record_cache_summary</td>
<td>recordcache_mb</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_record_cache_summary</td>
<td>logicalcache_mb</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_system_memory_summary</td>
<td>systemcache_mb</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_system_memory_summary</td>
<td>logicalsysmem_mb</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_transaction_summary</td>
<td>committed_txns</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>falcon_transaction_summary</td>
<td>roll_back_txns</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>global_privileges</td>
<td>user</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>grant_privileges</td>
<td>user_spec</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>inappropriate_slave_privileges</td>
<td>privileges</td>
<td>string</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Cardinality</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Collation</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Column_name</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Comment</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Index_type</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Key_name</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Non_unique</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Null</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Packed</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Seq_in_index</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Sub_part</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>index</td>
<td>Table</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>innodb_active_transactions</td>
<td></td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodb_blocked_transactions</td>
<td></td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodb_compression_time</td>
<td></td>
<td>long_counter</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>mysql</td>
<td>innodb_compression_time</td>
<td>innodb_compression_time</td>
<td>long_counter</td>
</tr>
<tr>
<td>mysql</td>
<td>innodb_lock_wait_transactions</td>
<td>innodb_lock_wait_transactions</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodb_plugins</td>
<td>innodb_plugin_list</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_add_alloc</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_created_per_sec</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_db_pages</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_dictionary_alloc</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_free_buffers</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_hit_rate</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_io_cur_pages</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_io_sum_pages</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_io_unzip_cur_pages</td>
<td>long</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_io_unzip_sum_pages</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_lru_len</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_modified_pages</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_not_young_hit_rate</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_old_db_pages</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_created</td>
<td>long_counter</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_not_young</td>
<td>long_counter</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_not_young_hit_rate</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_read</td>
<td>long_counter</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_read_ahead</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_written</td>
<td>long_counter</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_young</td>
<td>long_counter</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pages_young_per_sec</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pending_pages</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pending_writes_filling_list</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pending_writes_lru_len</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_pending_writes_single_page</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_reads_per_sec</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_size</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_total_alloc</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_unzip_lru_len</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_written_per_sec</td>
<td>double</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>----------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bp_young_hit_rate</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_bytes_per_read</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_datetime</td>
<td>string</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_hash_node_heap</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_hash_searches_per_sec_double</td>
<td>double</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_hash_table_size</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_hash_used_cells</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_ibuf_free_list_len</td>
<td>long</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_ibuf_inserts</td>
<td>long_counter</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_ibuf_merged_recs</td>
<td>long_counter</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_ibuf_merges</td>
<td>long_counter</td>
</tr>
</tbody>
</table>

309
<table>
<thead>
<tr>
<th>Namespace</th>
<th>Namespace Type</th>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_ibuf_seg_size</td>
<td>long</td>
<td>The Insert Buffer index header and tree size, in 16Kb page</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_ibuf_size</td>
<td>long</td>
<td>The Insert Buffer index tree size, in 16Kb pages</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_ibuf_logs</td>
<td>long</td>
<td>The number of pending log I/Os</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_ibuf_reads</td>
<td>long</td>
<td>The number of pending change buffer reads</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_ibuf_syncs</td>
<td>long</td>
<td>The number of pending sync operations</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_os_file_fsyncs</td>
<td>long_counter</td>
<td>The number of OS fsyncs.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_os_file_reads</td>
<td>long_counter</td>
<td>The number of OS file reads.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_os_file_writes</td>
<td>long_counter</td>
<td>The number of OS file writes.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_pending_flush_bp</td>
<td>long</td>
<td>The number of pending buffer pool flush operations.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_pending_flush_log</td>
<td>long</td>
<td>The number of pending log flush operations.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_pending_reads</td>
<td>long</td>
<td>The number of I/O pending reads.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_pending_writes</td>
<td>long</td>
<td>The number of I/O pending writes.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_io_syncs_per_sec</td>
<td>double</td>
<td>The number of fsyncs() per second.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_checkpoint_file</td>
<td>long</td>
<td>The log file number the last</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_checkpoint_lsn</td>
<td>long</td>
<td>The log sequence number of the last checkpoint.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_checkpoint_lsn64</td>
<td>long</td>
<td>The 64-bit variant of innodb_log_checkpoint_lsn (2.3.10+).</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_flushed_file</td>
<td>long</td>
<td>The log file number the last checkpoint was performed on.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_flushed_lsn</td>
<td>long</td>
<td>The point up to which the log was last flushed.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_flushed_lsn64</td>
<td>long</td>
<td>The 64-bit variant of innodb_log_flushed_lsn (2.3.10+).</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_io_per_sec</td>
<td>double</td>
<td>The number of log I/Os per second.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_io_total</td>
<td>long_counter</td>
<td>The total number of log I/Os.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_pending_checkpoin</td>
<td>long</td>
<td>The log pending checkpoint writes.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_pending_log_writes</td>
<td>long</td>
<td>The log pending log writes.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_sequence_file</td>
<td>long</td>
<td>The log sequence file number.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_sequence_lsn</td>
<td>long</td>
<td>The log sequence number.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_log_sequence_lsn64</td>
<td>long</td>
<td>The 64-bit variant of innodb_log_sequence_lsn (2.3.10+).</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_main_thd_log_flush_writes</td>
<td>long_counter</td>
<td>Count of log flushes</td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_main_thd_loops_background</td>
<td>long_counter</td>
<td>Count of background loops the main thread has done</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_main_thd_loops_flush</td>
<td>long_counter</td>
<td>Count of flush loops the main thread has done</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_main_thd_loops_one_sec</td>
<td>long_counter</td>
<td>Count of one second loops the main thread has done</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_main_thd_loops_sleep</td>
<td>long_counter</td>
<td>Count of sleeps the main thread has done inside the one second loop</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_main_thd_loops_ten_sec</td>
<td>long_counter</td>
<td>Count of ten second loops the main thread has done</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_non_hash_searches_per_sec</td>
<td>double</td>
<td>The number of nonadaptive hash index searches.</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_per_sec_avg</td>
<td>double</td>
<td>The number of seconds the averages for SHOW INNODB STATUS were calculated from.</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_reads_per_sec</td>
<td>double</td>
<td>The number of reads per second.</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_row_queries_inside</td>
<td>long</td>
<td>The number of queries executing inside InnoDB.</td>
<td></td>
</tr>
<tr>
<td>mysql innodbstatus</td>
<td>innodb_row_queries_queue</td>
<td>long</td>
<td>The number of queries in the queue,</td>
<td></td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_row_state</td>
<td>string</td>
<td>The current state of the main InnoDB thread.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_deleted</td>
<td>long_counter</td>
<td>The number of rows deleted from InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_deleted_per_sec</td>
<td>double</td>
<td>The number of rows deleted per second from InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_inserted</td>
<td>long_counter</td>
<td>The number of rows inserted into InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_inserted_per_sec</td>
<td>double</td>
<td>The number of rows inserted per second into InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_read</td>
<td>long_counter</td>
<td>The number of rows read from InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_read_per_sec</td>
<td>double</td>
<td>The number of rows read per second from InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_updated</td>
<td>long_counter</td>
<td>The number of rows updated in InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_rows_updated_per_sec</td>
<td>double</td>
<td>The number of rows updated per second in InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_mutex_os_waits</td>
<td>long_counter</td>
<td>The number of semaphore/mutex waits</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_mutex_rounds</td>
<td>long_counter</td>
<td>The number of semaphore/mutex round for the internal sync array.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_mutex_spin_wait</td>
<td>long_counter</td>
<td>The number of semaphore/mutex spin waits for the internal sync array.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_os_reservation_count</td>
<td>long_counter</td>
<td>The number of times a mutex wait was added to the internal sync array.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_os_signal_count</td>
<td>long_counter</td>
<td>Performance.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_rwlock_os_wait</td>
<td>long_counter</td>
<td>The number of exclusive (write) semaphore waits yielded to the OS.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_rwlock_rounds</td>
<td>long_counter</td>
<td>Performance.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_rwlock_spins</td>
<td>long_counter</td>
<td>The number of shared (read) semaphore spin waits within the sync array.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_rwlock_counter</td>
<td>long_counter</td>
<td>Performance.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_rwlock_spins_counter</td>
<td>long_counter</td>
<td>Ratio of innodb_sem_rwlock_rounds to innodb_sem_rwlock_spins.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_rwlock_counter</td>
<td>long_counter</td>
<td>Ratio of RW-Exclusive spin rounds to innodb_sem_rwlock_spins.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>-----------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_sem_spins_per_wait_rwlock_shared</td>
<td>double</td>
<td>Ratio of RW-Shared spin rounds to innodb_sem_shared_spins</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_trx_history_list_length</td>
<td>long</td>
<td>The transaction history list length.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_trx_id_counter</td>
<td>long_counter</td>
<td>Decoded decimal representation of the hex transaction ID counter in the latest InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_trx_id_counter1</td>
<td>long</td>
<td>The transaction counter roll over variable.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_trx_id_counter2</td>
<td>long</td>
<td>The main transaction count.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_trx_id_counter_str</td>
<td>string</td>
<td>The hex string based transaction ID counter in the latest InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_trx_purge_done_trx2</td>
<td>long</td>
<td>The transaction count that is already purged.</td>
</tr>
<tr>
<td>mysql</td>
<td>innodbstatus</td>
<td>innodb_trx_total_lock_structs</td>
<td>long</td>
<td>The number of InnoDB writes per second.</td>
</tr>
<tr>
<td>mysql</td>
<td>locked_processes</td>
<td>num_locked</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>locked_processes</td>
<td>long_running_time</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>locked_processes</td>
<td>num_long_running</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>locked_processes</td>
<td>long_running_time</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>locked_processes</td>
<td>num_long_running</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>masterlogs</td>
<td>File_size</td>
<td>long</td>
<td>The size of a binary log file.</td>
</tr>
<tr>
<td>mysql</td>
<td>masterlogs</td>
<td>filecount</td>
<td>long</td>
<td>The number of binary log files.</td>
</tr>
<tr>
<td>mysql</td>
<td>masterlogs</td>
<td>filesizesum</td>
<td>long</td>
<td>The total size of all the</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>---------------</td>
<td>---------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>masterlogs</td>
<td>Log_name</td>
<td>string</td>
<td>The name of the binary log file.</td>
</tr>
<tr>
<td>mysql</td>
<td>MasterStatus</td>
<td>Binlog_Do_DB</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>MasterStatus</td>
<td>Binlog_Ignore_DB</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>MasterStatus</td>
<td>File</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>MasterStatus</td>
<td>Position</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>backup_destination</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>collected_ts</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>collected_ts_counter</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>end_time_raw</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>end_time_ts</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>exit_state</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>last_error</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>last_error_code</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>lock_time</td>
<td>float</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>mysql_data_dir</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>progress_log</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>start_time_now</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>start_time_ts</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_history</td>
<td>total_time</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_backup_success_history</td>
<td>load_full_backup</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_incremental</td>
<td>total Increments</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_timing</td>
<td>lock_time</td>
<td>float</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>meb_timing</td>
<td>total_time</td>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>no_password</td>
<td>user</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>no_root_password</td>
<td>no_password</td>
<td>long</td>
<td>The type of command the thread is executing.</td>
</tr>
<tr>
<td>mysql</td>
<td>num_waiting_txns</td>
<td>num_waiting_txns</td>
<td>long</td>
<td>The default database, if one is selected.</td>
</tr>
<tr>
<td>mysql</td>
<td>prepared_count</td>
<td>prepared_count</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>privileges_on_all_dbs</td>
<td>user_spec</td>
<td>string</td>
<td>The host name of the client issuing the statement.</td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>Command</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>db</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>Host</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>Id</td>
<td>long</td>
<td>The connection identifier.</td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>Info</td>
<td>string</td>
<td>The statement that the thread is executing.</td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>State</td>
<td>string</td>
<td>An action, event, or state that indicates what the thread is doing.</td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>Time</td>
<td>long</td>
<td>The time in seconds that the thread has been in its current state.</td>
</tr>
<tr>
<td>mysql</td>
<td>processlist</td>
<td>User</td>
<td>string</td>
<td>The MySQL user who issued the statement.</td>
</tr>
<tr>
<td>mysql</td>
<td>repl_slave_privileges</td>
<td>repl_slaves</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>rollback_count</td>
<td>rollback_count</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>root_account_exists</td>
<td>root_account</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>root_remote_login</td>
<td>remote_login</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>blackout</td>
<td>long</td>
<td>Whether the server is connected.</td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>displayname</td>
<td>string</td>
<td>The display name of the server in the Dashboard.</td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>registration-complete</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>repl.group</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>repl.groupName</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>server.connected</td>
<td>long</td>
<td>Whether the server is connected.</td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>server.last_errno</td>
<td>string</td>
<td>The last MySQL server error message.</td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>server.last_error</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>server.reachable</td>
<td>long</td>
<td>Whether the server is reachable.</td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>server.version_numeric</td>
<td>long</td>
<td>The MySQL server version number.</td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>Time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------</td>
<td>-------------------------------</td>
<td>--------</td>
<td>-------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>transport</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Server</td>
<td>visible.displayname</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>server_admin_privileges</td>
<td>user_spec</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Binlog_Do_DB</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Binlog_Ignore_DB</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Connect_Retry</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Exec_Master_Log_Pos</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>File</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Last_Errno</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Last_Error</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Last_IO_Errno</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Last_IO_Error</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Last_SQL_Errno</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Last_SQL_Error</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_Host</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_ip</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_Log_File</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_Port</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_SSL_Allowed</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_SSL_CA_File</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_SSL_CA_Path</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_SSL_Cert</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_SSL_CIPHER</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_SSL_Cipher</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_SSL_Verify_Server_Cert</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_User</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Master_uuid</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Position</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Read_Master_Log_Pos</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Relay_Log_File</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Relay_Log_Pos</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Relay_Log_Space</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Relay_Master_Log_File</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Replicate_Do_DB</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Replicate_Do_Table</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Replicate_Ignore_DB</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Replicate_Ignore_Table</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Replicate_Wild_Do_Table</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Replicate_Wild_Ignore_Table</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Seconds_Behind_Master</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Skip_Counter</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>--------------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Slave_IO_Running</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Slave_IO_State</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Slave_SQL_Running</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Until.Condition</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Until_Log_File</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>SlaveStatus</td>
<td>Until_Log_Pos</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>sp_with_select_star</td>
<td>routine</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>comment</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>connection_id</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>database</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>errors</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>exec_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>explain_plan</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>host_from</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>host_to</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>no_good_index_used</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>no_index_used</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>query_type</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>rows</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>source_location</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>text</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>user</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>Statement</td>
<td>warnings</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementAnalysisSupport</td>
<td>explain_min_exec_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementAnalysisSupport</td>
<td>example_examples</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementAnalysisSupport</td>
<td>explain_explain</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementAnalysisSupport</td>
<td>enabled</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementAnalysisSupport</td>
<td>frequency</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>avg_exec_time</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>bytes_rel</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>count</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>count_rel</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>database</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>errors</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>exec_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>exec_time_rel</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>max_bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>max_exec_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>max_rows</td>
<td>long</td>
<td></td>
</tr>
</tbody>
</table>

319
<table>
<thead>
<tr>
<th>Namespace</th>
<th>Namespace Type</th>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>min_bytes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>min_exec_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>min_rows</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>no_good_index_used</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>no_index_used</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>query_type</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>rows</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>rows_rel</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>text</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>text_hash</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>StatementSummary</td>
<td>warnings</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Aborted_clients</td>
<td>long_counter</td>
<td>Networking.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Aborted_connects</td>
<td>long_counter</td>
<td>The number of failed attempts to connect to the MySQL server.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Binlog_cache_disk_use</td>
<td>long_counter</td>
<td>General.Logging</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Binlog_cache_use</td>
<td>long_counter</td>
<td>The number of transactions that used the temporary binary log cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Bytes_received</td>
<td>long_counter</td>
<td>The number of bytes received from all clients.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Bytes_sent</td>
<td>long_counter</td>
<td>The number of bytes sent to all clients.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_admin_commands</td>
<td>long_counter</td>
<td>Count of admin commands.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_db</td>
<td>long_counter</td>
<td>Count of ALTER DATABASE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_db_upgrade</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_event</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_function</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_procedure</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_server</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_table</td>
<td>long_counter</td>
<td>Count of ALTER</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_alter_tablespace</td>
<td>long_counter</td>
<td>Count of TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_analyze</td>
<td>long_counter</td>
<td>Count of ANALYZE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_assign_to_keycache</td>
<td>long_counter</td>
<td>Count of BACKUP TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_begin</td>
<td>long_counter</td>
<td>Count of BEGIN statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_binlog</td>
<td>long_counter</td>
<td>Number of calls to stored procedures.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_call_procedure</td>
<td>long_counter</td>
<td>Count of CHANGE DATABASE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_change_db</td>
<td>long_counter</td>
<td>Count of CHANGE MASTER statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_change_master</td>
<td>long_counter</td>
<td>Count of CHECK statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_checksum</td>
<td>long_counter</td>
<td>Count of CHECKSUM statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_commit</td>
<td>long_counter</td>
<td>Count of COMMIT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_db</td>
<td>long_counter</td>
<td>Count of CREATE DATABASE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_event</td>
<td>long_counter</td>
<td>Count of CREATE FUNCTION statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_function</td>
<td>long_counter</td>
<td>Count of CREATE INDEX statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_index</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_procedure</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------</td>
<td>------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_server</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_table</td>
<td>long_counter</td>
<td>Count of CREATE TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_trigger</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_udf</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_user</td>
<td>long_counter</td>
<td>Count of CREATE USER statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_create_view</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_dealloc_sql</td>
<td>long_counter</td>
<td>Count of DEALLOCATE PREPARE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_delete</td>
<td>long_counter</td>
<td>Count of DELETE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_delete_multi</td>
<td>long_counter</td>
<td>Count of multi-table DELETE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_do</td>
<td>long_counter</td>
<td>Count of DO statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_db</td>
<td>long_counter</td>
<td>Count of DROP DATABASE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_event</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_function</td>
<td>long_counter</td>
<td>Count of DROP FUNCTION statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_index</td>
<td>long_counter</td>
<td>Count of DROP INDEX statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_procedure</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_server</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_table</td>
<td>long_counter</td>
<td>Count of DROP TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_trigger</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_user</td>
<td>long_counter</td>
<td>Count of DROP USER statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_drop_view</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_empty_query</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------</td>
<td>--------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_execute_sql</td>
<td>long_counter</td>
<td>Count of EXECUTE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_flush</td>
<td>long_counter</td>
<td>Count of FLUSH statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_grant</td>
<td>long_counter</td>
<td>Count of GRANT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_ha_close</td>
<td>long_counter</td>
<td>Count of HANDLER CLOSE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_ha_open</td>
<td>long_counter</td>
<td>Count of HANDLER OPEN statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_ha_read</td>
<td>long_counter</td>
<td>Count of HANDLER READ statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_help</td>
<td>long_counter</td>
<td>Count of HELP statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_insert</td>
<td>long_counter</td>
<td>Count of INSERT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_insert_select</td>
<td>long_counter</td>
<td>Count of INSERT SELECT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_install_plugin</td>
<td>long_counter</td>
<td>Count of KILL statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_kill</td>
<td>long_counter</td>
<td>Count of LOAD statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_load</td>
<td>long_counter</td>
<td>Count of LOAD MASTER DATA statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_load_master_data</td>
<td>long_counter</td>
<td>Count of LOAD MASTER TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_load_master_table</td>
<td>long_counter</td>
<td>Count of LOAD TABLES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_lock_tables</td>
<td>long_counter</td>
<td>Count of LOCK TABLES statements.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------</td>
<td>-----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_optimize</td>
<td>long_counter</td>
<td>Count of OPTIMIZE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_preload_keys</td>
<td>long_counter</td>
<td>Count of PRELOAD KEYS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_prepare_sql</td>
<td>long_counter</td>
<td>Count of PREPARE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_purge</td>
<td>long_counter</td>
<td>Count of PURGE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_purge_before_date</td>
<td>long_counter</td>
<td>Count of PURGE BEFORE DATE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_release_savepoint</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_rename_table</td>
<td>long_counter</td>
<td>Count of RENAME TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_rename_user</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_repair</td>
<td>long_counter</td>
<td>Count of REPAIR statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_replace</td>
<td>long_counter</td>
<td>Count of REPLACE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_replace_select</td>
<td>long_counter</td>
<td>Count of REPLACE SELECT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_reset</td>
<td>long_counter</td>
<td>Count of RESET statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_restore_table</td>
<td>long_counter</td>
<td>Count of RESTORE TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_revoke</td>
<td>long_counter</td>
<td>Count of REVOKE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_revoke_all</td>
<td>long_counter</td>
<td>Count of REVOKE ALL statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_rollback</td>
<td>long_counter</td>
<td>Count of ROLLBACK statements.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------------</td>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_rollback_to_savepoint</td>
<td>long_counter</td>
<td>Count of SAVEPOINT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_savpoint</td>
<td>long_counter</td>
<td>Count of SELECT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_select</td>
<td>long_counter</td>
<td>Count of SET OPTION statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_set_option</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_authors</td>
<td>long_counter</td>
<td>Count of SHOW statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_binlog_events</td>
<td>long_counter</td>
<td>Count of SHOW BINLOG EVENTS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_binlogs</td>
<td>long_counter</td>
<td>Count of SHOW BINLOGS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_charsets</td>
<td>long_counter</td>
<td>Count of SHOW CHARSETS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_collations</td>
<td>long_counter</td>
<td>Count of SHOW COLLATIONS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_column_types</td>
<td>long_counter</td>
<td>Count of SHOW COLUMN TYPES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_contributors</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_create_db</td>
<td>long_counter</td>
<td>Count of SHOW CREATE DATABASE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_create_event</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_create_func</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_create_proc</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_create_table</td>
<td>long_counter</td>
<td>Count of SHOW CREATE TABLE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_create_trigger</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_databases</td>
<td>long_counter</td>
<td>Count of SHOW DATABASES statements.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>------------------------</td>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_engine_logs</td>
<td>long_counter</td>
<td>DATABASES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_engine_mutex</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_engine_status</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_errors</td>
<td>long_counter</td>
<td>Count of SHOW ERRORS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_events</td>
<td>long_counter</td>
<td>Count of SHOW EVENTS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_fields</td>
<td>long_counter</td>
<td>Count of SHOW FIELDS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_function_status</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_grants</td>
<td>long_counter</td>
<td>Count of SHOW GRANTS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_innodb_status</td>
<td>long_counter</td>
<td>Count of SHOW INNODB STATUS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_keys</td>
<td>long_counter</td>
<td>Count of SHOW KEYS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_logs</td>
<td>long_counter</td>
<td>Count of SHOW LOGS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_master_status</td>
<td>long_counter</td>
<td>Count of SHOW MASTER STATUS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_ndb_status</td>
<td>long_counter</td>
<td>Count of SHOW NDB STATUS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_new_master</td>
<td>long_counter</td>
<td>Count of SHOW NEW MASTER statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_open_tables</td>
<td>long_counter</td>
<td>Count of SHOW OPEN TABLES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_plugins</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_privileges</td>
<td>long_counter</td>
<td>Count of SHOW PRIVILEGES statements.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_procedure_status</td>
<td>long_counter</td>
<td>PRIVILEGES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_processlist</td>
<td>long_counter</td>
<td>Count of SHOW PROCESSELIST statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_profile</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_profiles</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_slave_hosts</td>
<td>long_counter</td>
<td>Count of SHOW SLAVE HOSTS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_slave_status</td>
<td>long_counter</td>
<td>Count of SHOW SLAVE STATUS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_status</td>
<td>long_counter</td>
<td>Count of SHOW STATUS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_storage_engines</td>
<td>long_counter</td>
<td>Count of SHOW STORAGE ENGINES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_table_status</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_tables</td>
<td>long_counter</td>
<td>Count of SHOW TABLES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_triggers</td>
<td>long_counter</td>
<td>Count of SHOW TRIGGERS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_variables</td>
<td>long_counter</td>
<td>Count of SHOW VARIABLES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_show_warnings</td>
<td>long_counter</td>
<td>Count of SHOW WARNINGS statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_slave_start</td>
<td>long_counter</td>
<td>Count of START SLAVE statements.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------------------</td>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_slave_stop</td>
<td>long_counter</td>
<td>Count of STOP SLAVE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_stmt_close</td>
<td>long_counter</td>
<td>Count of STATEMENT CLOSE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_stmt_execute</td>
<td>long_counter</td>
<td>Count of STATEMENT EXECUTE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_stmt_fetch</td>
<td>long_counter</td>
<td>Count of STATEMENT FETCH statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_stmt_prepare</td>
<td>long_counter</td>
<td>Count of STATEMENT PREPARE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_stmt_reprepare</td>
<td>long_counter</td>
<td>Count of STATEMENT RESET statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_stmt_send_long_data</td>
<td>long_counter</td>
<td>Count of STATEMENT SEND LONG DATA statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_truncate</td>
<td>long_counter</td>
<td>Count of TRUNCATE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_uninstall_plugin</td>
<td>long_counter</td>
<td>Count of UNLOCK TABLES statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_unlock_tables</td>
<td>long_counter</td>
<td>Count of UPDATE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_update</td>
<td>long_counter</td>
<td>Count of UPDATE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_update_multi</td>
<td>long_counter</td>
<td>Count of multi-table UPDATE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_xa_commit</td>
<td>long_counter</td>
<td>Count of XA COMMIT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_xa_end</td>
<td>long_counter</td>
<td>Count of XA END statements.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------------------</td>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_xa_prepare</td>
<td>long_counter</td>
<td>Count of XA PREPARE statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_xa_recover</td>
<td>long_counter</td>
<td>Count of XA RECOVER statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_xa_rollback</td>
<td>long_counter</td>
<td>Count of XA ROLLBACK statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Com_xa_start</td>
<td>long_counter</td>
<td>Count of XA START statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Compression</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Connections</td>
<td>long_counter</td>
<td>The number of connection attempts.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Created_tmp_disk_tables</td>
<td>long_counter</td>
<td>General.Temporary</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Created_tmp_files</td>
<td>long_counter</td>
<td>How many temporary files mysqld has created.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Created_tmp_tables</td>
<td>long_counter</td>
<td>How many temporary tables mysqld has created.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Delayed_errors</td>
<td>long_counter</td>
<td>Performance.Delayed</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Delayed_insert_threads</td>
<td>long</td>
<td>The number of INSERT DELAYED thread handlers in use. For non-transactional tables only.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Delayed_writes</td>
<td>long_counter</td>
<td>The number of INSERT DELAYED rows written. For non-transactional tables only.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>displayname</td>
<td>string</td>
<td>The display name of the server in the Dashboard.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Flush_commands</td>
<td>long_counter</td>
<td>The number of FLUSH statements executed.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>groupname</td>
<td>string</td>
<td>The name of the group</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_commit</td>
<td>long_counter</td>
<td>The number of internal COMMIT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_delete</td>
<td>long_counter</td>
<td>The number of times that rows have been deleted from tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_discover</td>
<td>long_counter</td>
<td>The number of times that tables have been discovered.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_prepare</td>
<td>long_counter</td>
<td>A counter for the prepare phase of two-phase commit operations.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_read_first</td>
<td>long_counter</td>
<td>The number of times the first entry was read from an index.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_read_key</td>
<td>long_counter</td>
<td>The number of requests to read a row based on a key.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_read_next</td>
<td>long_counter</td>
<td>The number of requests to read the next row in key order.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_read_prev</td>
<td>long_counter</td>
<td>The number of requests to read the previous row in key order.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_read_rnd</td>
<td>long_counter</td>
<td>The number of requests to read a row based on a fixed position.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_read_rnd_next</td>
<td>long_counter</td>
<td>The number of requests to read the next row in the data file.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_rollback</td>
<td>long_counter</td>
<td>Miscellaneous.Handler</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_savepoint</td>
<td>long_counter</td>
<td>The number of requests for a storage engine to place a savepoint.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_savepoint_rollback</td>
<td>long_counter</td>
<td>The number of requests for a storage engine to roll back to a savepoint.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_update</td>
<td>long_counter</td>
<td>The number of requests to update a row in a table.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Handler_write</td>
<td>long_counter</td>
<td>The number of requests to insert a row in a table.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_pages_data</td>
<td>long</td>
<td>The number of pages containing data (dirty or clean)</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_pages_dirty</td>
<td>long</td>
<td>The number of pages currently dirty in the InnoDB buffer pool.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_pages_flushed</td>
<td>long_counter</td>
<td>The number of page flush requests in the InnoDB buffer pool.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_pages_free</td>
<td>long</td>
<td>The number of InnoDB buffer pool pages free.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_pages_latched</td>
<td>long</td>
<td>The number of latched pages in InnoDB buffer pool.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_pages_mixed</td>
<td></td>
<td>Performance.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_pages_tot</td>
<td></td>
<td>The total size of the InnoDB buffer pool, in pages.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_buffer_pool_read_ahead</td>
<td>long_counter</td>
<td>The number of random</td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_buffer_pool_read_aheads_seq</td>
<td>long_counter</td>
<td>The number of sequential read-aheads initiated by InnoDB.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_buffer_pool_read_requests</td>
<td>long_counter</td>
<td>The number of logical read requests InnoDB has done.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_buffer_pool_reads</td>
<td>long_counter</td>
<td>Performance. InnoDB</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_buffer_pool_wait_free</td>
<td>long_counter</td>
<td>Number of waits for pages to be flushed.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_buffer_pool_write_requests</td>
<td>long_counter</td>
<td>The number of writes done to the InnoDB buffer pool.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_fsyncs</td>
<td>long_counter</td>
<td>The number of fsync() operations so far.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_pending_fsyncs</td>
<td>long</td>
<td>The current number of pending fsync() operations.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_pending_reads</td>
<td>long</td>
<td>The current number of pending reads.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_pending_writes</td>
<td>long</td>
<td>The number of pending writes.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_read</td>
<td>long_counter</td>
<td>The amount of data read so far, in bytes.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_reads</td>
<td>long_counter</td>
<td>The total number of data reads.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_writes</td>
<td>long_counter</td>
<td>The total number of data writes.</td>
<td></td>
</tr>
<tr>
<td>mysql status</td>
<td>Innodb_data_written</td>
<td>long_counter</td>
<td>The amount of data written in bytes.</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_dblwr_pages_written</td>
<td>long_counter</td>
<td>The number of doublewrite pages that have been written.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_dblwr_writes</td>
<td>long_counter</td>
<td>The number of doublewrite operations that have been performed.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_log_waits</td>
<td>long_counter</td>
<td>Performance.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_log_write_requests</td>
<td>long_counter</td>
<td>The number of log write requests.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_log_writes</td>
<td>long_counter</td>
<td>The number of physical writes to the log.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_os_log_fsyncs</td>
<td>long_counter</td>
<td>The number of fsync() writes done to the log file.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_os_log_pending_fsyncs</td>
<td>long_counter</td>
<td>The number of pending log file fsync() operations.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_os_log_pending_writes</td>
<td>long</td>
<td>The number of pending log file writes.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_os_log_written</td>
<td>long_counter</td>
<td>The number of bytes written to the log file.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_page_size</td>
<td>long</td>
<td>The compiled-in InnoDB page size.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_pages_created</td>
<td>long_counter</td>
<td>The number of pages created.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_pages_read</td>
<td>long_counter</td>
<td>The number of pages read.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_pages_written</td>
<td>long_counter</td>
<td>The number of pages written.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_row_lock_current_waits</td>
<td>long</td>
<td>The number of row locks currently being waited for.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_row_lock_time</td>
<td>long_counter</td>
<td>The total time spent in acquiring row locks, in milliseconds.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_row_lock_time_avg</td>
<td>long</td>
<td>The average time to acquire a row lock, in milliseconds.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_row_lock_time_max</td>
<td>long</td>
<td>The maximum time to acquire a row lock, in milliseconds.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_row_lock_waits</td>
<td>long_counter</td>
<td>The number of times a row lock had to be waited for.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_rows_deleted</td>
<td>long_counter</td>
<td>The number of rows deleted from InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_rowsInserted</td>
<td>long_counter</td>
<td>The number of rows inserted into InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_rows_read</td>
<td>long_counter</td>
<td>The number of rows read from InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Innodb_rows_updated</td>
<td>long_counter</td>
<td>The number of rows updated in InnoDB tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Key_blocks_not_flushed</td>
<td>long</td>
<td>Performance.Keys</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Key_blocks_unused</td>
<td>long</td>
<td>The number of unused blocks in the key cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Key_blocks_used</td>
<td>long</td>
<td>The number of used blocks in the key cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Key_read_requests</td>
<td>long_counter</td>
<td>The number of requests to read a key block from the cache.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Key_reads</td>
<td>long_counter</td>
<td>The number of physical reads of a key block from disk.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Key_write_requests</td>
<td>long_counter</td>
<td>The number of requests to write a key block to the cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Key_writes</td>
<td>long_counter</td>
<td>The number of physical writes of a key block to disk.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Last_query_cost</td>
<td>long</td>
<td>Performance.Cost</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Max_used_connections</td>
<td>long</td>
<td>Networking.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ndb_number_of_data_nodes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ndb_number_of_ready_data_nodes</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Not_flushed_delayed_rows</td>
<td>long</td>
<td>The number of rows waiting to be written in INSERT DELAY queues. For non-transactional tables only.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Open_files</td>
<td>long</td>
<td>The number of files that are open.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Open_streams</td>
<td>long</td>
<td>The number of streams that are open (used mainly for logging)</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Open_table_definitions</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Open_tables</td>
<td>long</td>
<td>The number of tables that are open.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Opened_files</td>
<td>long_counter</td>
<td>The number of tables that have been opened.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Opened_table Definitions</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Opened_tables</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Prepared_stmt_count</td>
<td>long</td>
<td>The current number of prepared statements.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_free_blocks</td>
<td>long</td>
<td>The number of free memory blocks in the query cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_free_memory</td>
<td>long</td>
<td>The amount of free memory for the query cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_hits</td>
<td>long_counter</td>
<td>The number of query cache hits.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_inserts</td>
<td>long_counter</td>
<td>The number of query cache inserts.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_lowmem_prunes</td>
<td>long_counter</td>
<td>Performance.Query Cache</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_notCached</td>
<td>long_counter</td>
<td>Performance.Query Cache</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_queries_in_cache</td>
<td>long</td>
<td>The number of queries registered in the query cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Qcache_total_blocks</td>
<td>long</td>
<td>The total number of blocks in the query cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Queries</td>
<td>long_counter</td>
<td>The number of statements that clients have sent to the server.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Questions</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Rpl_status</td>
<td>long</td>
<td>The status of fail-safe replication (not yet implemented)</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Select_full_join</td>
<td>long_counter</td>
<td>Performance.Selects</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Select_full_range_join</td>
<td>long_counter</td>
<td>The number of joins that used a range search on a reference table.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Select_range</td>
<td>long_counter</td>
<td>The number of joins that used ranges</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Select_range_check</td>
<td>long_counter</td>
<td>The number of joins without keys that check for key usage after each row.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Select_scan</td>
<td>long_counter</td>
<td>The number of joins that did a full scan of the first table.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Slave_open_temp_tables</td>
<td>long</td>
<td>Networking.Replication</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Slave_retried_transactions</td>
<td>long_counter</td>
<td>Networking.Replication</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Slave_running</td>
<td>string</td>
<td>This is ON if this server is a slave that is connected to a master.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Slow_launch_threads</td>
<td>long_counter</td>
<td>Performance.Slow Items</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Slow_queries</td>
<td>long_counter</td>
<td>The number of queries that have taken more than long_query_time seconds.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Sort_merge_passes</td>
<td>long_counter</td>
<td>The number of merge passes that the sort algorithm has had to do.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Sort_range</td>
<td>long_counter</td>
<td>The number of sorts that were done using ranges.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Sort_rows</td>
<td>long_counter</td>
<td>The number of sorted rows.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Sort_scan</td>
<td>long_counter</td>
<td>The number of sorts that were done by scanning the table.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_accept_renegotiates</td>
<td>long_counter</td>
<td>The number of renegotiates needed to</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------</td>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_accepts</td>
<td>long_counter</td>
<td>The number of attempted SSL connections.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_callback_cache_hits</td>
<td>long_counter</td>
<td>The number of callback cache hits.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_cipher</td>
<td>string</td>
<td>SSL cipher to use (implies --ssl).</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_cipher_list</td>
<td>string</td>
<td>The list of SSL cipher strings.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_client_connects</td>
<td>long_counter</td>
<td>The number of attempted connections to an SSL-enabled master.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_connect_renegotiates</td>
<td>long_counter</td>
<td>Number of renegotiates needed to connect to an SSL-enabled master.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_ctx_verify_depth</td>
<td>long</td>
<td>The SSL context verification depth.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_ctx_verify_mode</td>
<td>long</td>
<td>The SSL certificate verification mode used by the server.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_default_timeout</td>
<td>long</td>
<td>The SSL default timeout.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_finished_accepts</td>
<td>long_counter</td>
<td>The number of successful SSL connections to the server.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl.finished_connects</td>
<td>long_counter</td>
<td>The number of successful SSL slave connections to the server.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_session_cache_hits</td>
<td>long_counter</td>
<td>The number of SSL connections to the server.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------</td>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_session_cache_misses</td>
<td>long_counter</td>
<td>The number of SSL session cache misses.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_session_cache_mode</td>
<td>string</td>
<td>The SSL session cache mode.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_session_cache_overflow</td>
<td>long_counter</td>
<td>The number of SSL cache overflows.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_session_cache_size</td>
<td>long</td>
<td>The size of the SSL session cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_session_cache_timeouts</td>
<td>long_counter</td>
<td>The number of session cache timeouts.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_sessions_reused</td>
<td>long</td>
<td>The number of SSL sessions from the cache that were reused.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_used_session_cache_entries</td>
<td>string</td>
<td>The number of SSL session cache entries used.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_verify_depth</td>
<td>long</td>
<td>Depth of verification for replication SSL connections.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_verify_mode</td>
<td>long</td>
<td>Mode of verification for replication SSL connections.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Ssl_version</td>
<td>string</td>
<td>The SSL version number.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Table_locks_immediate</td>
<td>long_counter</td>
<td>The number of times that a table lock was acquired immediately.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Table_locks_waited</td>
<td>long_counter</td>
<td>Performance.Locks</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Tc_log_max_pages_used</td>
<td>long</td>
<td>General.Logging</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Tc_log_page_size</td>
<td>long</td>
<td>General.Logging</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------------</td>
<td>-----------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Tc_log_page_waits</td>
<td>long</td>
<td>General.Logging</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Threads_cached</td>
<td>long</td>
<td>The number of threads in the thread cache.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Threads_connected</td>
<td>long</td>
<td>The number of currently open connections.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Threads_created</td>
<td>long_counter</td>
<td>The number of threads created to handle connections.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Threads_running</td>
<td>long</td>
<td>The number of threads that are not sleeping.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Uptime</td>
<td>long</td>
<td>The number of seconds the server has been up.</td>
</tr>
<tr>
<td>mysql</td>
<td>status</td>
<td>Uptime_since_flush_status</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>strong_privileges</td>
<td>user</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>num_rows</td>
<td>long</td>
<td>MySQL Network.Table Data</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_auto_increment</td>
<td>long</td>
<td>The next AUTO_INCREMENT value.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_avg_row_length</td>
<td>long</td>
<td>The average row length.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_collation</td>
<td>string</td>
<td>The table's character set and collation.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_comment</td>
<td>string</td>
<td>The table comment.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_create_time</td>
<td>string</td>
<td>When the data file was created.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_data_free</td>
<td>long</td>
<td>The number of allocated but unused bytes.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_data_length</td>
<td>long</td>
<td>The length of the data file.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_engine</td>
<td>string</td>
<td>The storage engine used by a table.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>-----------------------------</td>
<td>-------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_index_length</td>
<td>long</td>
<td>The length of the index file.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_max_data_length</td>
<td>long</td>
<td>The maximum length of the data file.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_name</td>
<td>string</td>
<td>The name of a table.</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_row_format</td>
<td>string</td>
<td>The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact)</td>
</tr>
<tr>
<td>mysql</td>
<td>table</td>
<td>table_version</td>
<td>long</td>
<td>The version number of the table’s .frm file.</td>
</tr>
<tr>
<td>mysql</td>
<td>tables_no_myisam_index_stats</td>
<td>table_list</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>tables_no_unique_keys</td>
<td>table_list</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Auto_increment</td>
<td>long</td>
<td>The next AUTO_INCREMENT value.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Avg_row_length</td>
<td>long</td>
<td>The average row length.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Check_time</td>
<td>string</td>
<td>When the table was last checked.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Checksum</td>
<td>string</td>
<td>The live checksum value (if any)</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Collation</td>
<td>string</td>
<td>The table’s character set.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Comment</td>
<td>string</td>
<td>The comment used when creating the table.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Create_options</td>
<td>string</td>
<td>Extra options used with CREATE TABLE</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Create_time</td>
<td>string</td>
<td>When the table was created.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Data_free</td>
<td>long</td>
<td>The number of allocated but unused bytes.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Data_length</td>
<td>long</td>
<td>The length of the data file.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
<td>-----------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Engine</td>
<td>string</td>
<td>The storage engine for the table.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Index_length</td>
<td>long</td>
<td>The length of the index file.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Max_data_length</td>
<td>long</td>
<td>The maximum length of the data file.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Name</td>
<td>string</td>
<td>The table name.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Row_format</td>
<td>string</td>
<td>The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact).</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Rows</td>
<td>long</td>
<td>The number of rows in the table.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Update_time</td>
<td>string</td>
<td>When the data file was last updated.</td>
</tr>
<tr>
<td>mysql</td>
<td>tablestatus</td>
<td>Version</td>
<td>long</td>
<td>The version number of the table's .frm file.</td>
</tr>
<tr>
<td>mysql</td>
<td>test_database</td>
<td>Database (test)</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>transactions_in_serial_log</td>
<td>transactions_in_serial_log</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>trigger_with_select_star</td>
<td>trigger_defn</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>user_on_missing_db</td>
<td>db_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>user_on_missing_db</td>
<td>user</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>user_on_missing_table</td>
<td>user</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>auto_increment_increment</td>
<td>long</td>
<td>Auto-increment columns are incremented by this value.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>auto_increment_offset</td>
<td>long</td>
<td>Offset added to auto-increment columns.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>autocommit</td>
<td>string</td>
<td>Creating and dropping stored procedures alters ACLs.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>automatic_sp_privileges</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>back_log</td>
<td>long</td>
<td>The number of outstanding connection requests MySQL can have.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>basedir</td>
<td>string</td>
<td>General.Directories / Files</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>big_tables</td>
<td>string</td>
<td>Memory.Caches</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>binlog_cache_size</td>
<td>long</td>
<td>Memory.Caches</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>binlog_direct_non_transactional_updates</td>
<td>string</td>
<td>Memory.Caches</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>binlog_format</td>
<td>string</td>
<td>Memory.Caches</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>bulk_insert_buffer_size</td>
<td>long</td>
<td>Size of tree cache used in bulk insert optimization.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_set_client</td>
<td>string</td>
<td>Current client character set.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_set_connection</td>
<td>string</td>
<td>Current connection character set.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_set_database</td>
<td>string</td>
<td>The character set used by the default database.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_set_filesystem</td>
<td>string</td>
<td>Set the file system character set.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_set_results</td>
<td>string</td>
<td>Current result character set.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_set_server</td>
<td>string</td>
<td>SQL.Charsets</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_set_system</td>
<td>string</td>
<td>The character set used by the server for storing identifiers.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>character_sets_dir</td>
<td>string</td>
<td>Directory where character sets are.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>collation_connection</td>
<td>string</td>
<td>The collation of the connection.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>collation_database</td>
<td>string</td>
<td>The collation used by the default database.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>collation_server</td>
<td>string</td>
<td>Set the default collation.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>--------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>completion_type</td>
<td>long</td>
<td>Default completion type.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>concurrent_insert</td>
<td>long</td>
<td>Use concurrent insert with MyISAM.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>connect_timeout</td>
<td>long</td>
<td>Connections.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>datadir</td>
<td>string</td>
<td>Path to the database root.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>date_format</td>
<td>string</td>
<td>The DATE format (For future)</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>datetime_format</td>
<td>string</td>
<td>The DATETIME/TIMESTAMP format (for future)</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>default_week_format</td>
<td>long</td>
<td>The default week format used by WEEK() functions.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>delay_key_write</td>
<td>string</td>
<td>Type of DELAY_KEY_WRITE.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>delayed_insert_limit</td>
<td>long</td>
<td>General.Performance</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>delayed_insert_timeout</td>
<td>long</td>
<td>General.Performance</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>delayed_queue_size</td>
<td>long</td>
<td>General.Performance</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>div_precision_increment</td>
<td>long</td>
<td>SQL.Formats</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>engine_condition_pushdown</td>
<td>string</td>
<td>Push supported query conditions to the storage engine.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>error_count</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>event_scheduler</td>
<td>string</td>
<td>General.Miscellaneous</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>expire_logs_days</td>
<td>long</td>
<td>Flush tables to disk between SQL commands.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>flush</td>
<td>string</td>
<td>A dedicated thread is created to flush all tables at the given interval.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>flush_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>foreign_key_checks</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>ft_boolean_syntax</td>
<td>string</td>
<td>List of operators for MATCH ... AGAINST ( ... IN BOOLEAN MODE).</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>ft_max_word_len</td>
<td>long</td>
<td>SQL.Full Text Search</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>ft_min_word_len</td>
<td>long</td>
<td>SQL.Full Text Search</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>ft_query_expansion_limit</td>
<td>long</td>
<td>Number of best matches to use for query expansion.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>ft_stopword_file</td>
<td>string</td>
<td>Use stopwords from this file instead of built-in list.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>general_log</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>general_log_file</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>group_concat_max_len</td>
<td>long</td>
<td>The maximum length of the result of function group_concat.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_archive</td>
<td>string</td>
<td>Whether mysqld supports archive tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_bdb</td>
<td>string</td>
<td>Is Berkeley DB supported.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_blackhole_engine</td>
<td>string</td>
<td>Whether mysqld supports BLACKHOLE tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_community_features</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_compress</td>
<td>string</td>
<td>Availability of the zlib compression library.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_crypt</td>
<td>string</td>
<td>Availability of the crypt() system call.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_csv</td>
<td>string</td>
<td>Whether mysqld supports csv tables.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_dynamic_loading</td>
<td>string</td>
<td>Whether mysqld supports dynamic loading of plugins.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_example_engine</td>
<td>string</td>
<td>Whether mysqld supports EXAMPLE tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_federated_engine</td>
<td>string</td>
<td>Whether mysqld supports FEDERATED tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_geometry</td>
<td>string</td>
<td>Whether mysqld supports spatial data types.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_innodb</td>
<td>string</td>
<td>Whether mysqld supports InnoDB tables. No longer used as of MEM 2.3.13.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_isam</td>
<td>string</td>
<td>Whether mysqld supports isam tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_merge_engine</td>
<td>string</td>
<td>Whether mysqld supports merge tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_ndbcluster</td>
<td>string</td>
<td>Whether mysqld supports NDB Cluster tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_openssl</td>
<td>string</td>
<td>Whether mysqld supports SSL connections.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_partitioning</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_profiling</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_query_cache</td>
<td>string</td>
<td>Whether mysqld supports query cache.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_raid</td>
<td>string</td>
<td>Whether mysqld supports the RAID option.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_rtree_keys</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_ssl</td>
<td>string</td>
<td>Whether the server supports an SSL connection.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>have_symlink</td>
<td>string</td>
<td>Is symbolic link support enabled.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>hostname</td>
<td>string</td>
<td>The name of the server host.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>identity</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>ignore_builtin_innodb</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>init_connect</td>
<td>string</td>
<td>Command(s) that are executed for each new connection.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>init_file</td>
<td>string</td>
<td>Read SQL commands from this file at startup.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>init_slave</td>
<td>string</td>
<td>Command(s) that are executed when a slave connects to a master.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_adaptive_flushing</td>
<td>string</td>
<td>Memory.Buffers</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_adaptive_hash_index</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_additional_mem_pool</td>
<td>long</td>
<td>Amount by which InnoDB auto-extends the data files for a tablespace, in megabytes.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_autoextend_increment</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_autoinc_lock_mode</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_buffer_pool_aware_mem</td>
<td>long</td>
<td>Memory.Buffers</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_buffer_pool_size</td>
<td>long</td>
<td>Memory.Buffers</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_change_buffering</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_checksums</td>
<td>string</td>
<td>Enable InnoDB</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_commit_concurrency</td>
<td>long</td>
<td>Helps in performance tuning in heavily concurrent environments.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_concurrency_tickets</td>
<td>long</td>
<td>Table Types.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_data_file_path</td>
<td>string</td>
<td>Path to individual files and their sizes.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_data_home_dir</td>
<td>string</td>
<td>The common location for InnoDB tablespaces</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_doublewrite</td>
<td>string</td>
<td>Enable InnoDB doublewrite buffer</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_fast_shutdown</td>
<td>long</td>
<td>Speeds up the shutdown process of the InnoDB storage engine by deferring cleanup operations until the server restarts.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_file_format</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_file_format_check</td>
<td>string</td>
<td>Number of file I/O threads in InnoDB.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_file_io_threads</td>
<td>long</td>
<td>Stores each InnoDB table and associated indexes in an .ibd file in the database directory.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_file_per_table</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_flush_log_at_trx_commit</td>
<td>string</td>
<td>Table Types.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_flush_method</td>
<td>string</td>
<td>With which method to flush data.</td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_force_recovery</td>
<td>long</td>
<td>Table Types.InnoDB</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_io_capacity</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_lock_wait_timeout</td>
<td>long</td>
<td>Table Types.InnoDB</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_locks_unsafe_for_binlog</td>
<td>string</td>
<td>Table Types.InnoDB</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_log_arch_dir</td>
<td>string</td>
<td>Where full logs should be archived.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_log_archive</td>
<td>string</td>
<td>Set to 1 to have logs archived.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_log_buffer_size</td>
<td>long</td>
<td>General.Logs</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_log_file_size</td>
<td>long</td>
<td>Size of each log file in a log group.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_log_files_in_group</td>
<td>long</td>
<td>General.Logs</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_log_group_home_dir</td>
<td>string</td>
<td>Path to InnoDB log files.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_max_dirty_pages_pct</td>
<td>long</td>
<td>Percentage of dirty pages allowed in InnoDB buffer pool.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_max_purge_lag</td>
<td>long</td>
<td>Desired maximum length of the purge queue (0 = no limit)</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_mirrored_log_groups</td>
<td>long</td>
<td>Number of identical copies of log groups to keep for the database.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_old_blocks_pct</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_old_blocks_time</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_open_files</td>
<td>long</td>
<td>The maximum number of files that InnoDB keeps open at the same time.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_read_ahead_threshold</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_read_io_threads</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_rollback_on_timeout</td>
<td>string</td>
<td>Unknown.Unknown</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>inndodb_spin_wait_delay</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_stats_on_metadata</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_stats_sample_pages</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_strict_mode</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_support_xa</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>variables</td>
<td>innodb_sync_spin_loops</td>
<td>long</td>
<td>Count of spin-loop rounds in InnoDB mutexes</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_table_locks</td>
<td>string</td>
<td>Enable InnoDB locking in LOCK TABLES</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_thread_concurrency</td>
<td>long</td>
<td>Table Types.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_thread_sleep_delay</td>
<td>long</td>
<td>Table Types.InnoDB</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_use_legacy_cardinality_algorithm</td>
<td>string</td>
<td>Applies only to MySQL 5.1 and earlier.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_use_sys_malloc</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>innodb_write_io_threads</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>insert_id</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>interactive_timeout</td>
<td>long</td>
<td>Connections.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>join_buffer_size</td>
<td>long</td>
<td>The size of the buffer that is used for full joins.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>keep_files_on_create</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>key_buffer_size</td>
<td>long</td>
<td>The size of the buffer used for index blocks for MyISAM tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>key_cache_age_threshold</td>
<td>long</td>
<td>Memory.Caches</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>key_cache_block_size</td>
<td>long</td>
<td>The default size of key cache blocks.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>key_cache_division_limit</td>
<td>long</td>
<td>The minimum percentage of warm blocks in the key cache.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>language</td>
<td>string</td>
<td>Client error messages in given language. May be given as a full path.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>large_files_support</td>
<td>string</td>
<td>Whether large files are supported.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>large_page_size</td>
<td>long</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>large_pages</td>
<td>string</td>
<td>Enable support for large pages.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>last_insert_id</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>lc_time_names</td>
<td>string</td>
<td>General.Miscellaneous</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>license</td>
<td>string</td>
<td>Type of license for the server.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>local_infile</td>
<td>string</td>
<td>Enable/disable LOAD DATA LOCAL INFILE (takes values 1</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>locked_in_memory</td>
<td>string</td>
<td>Whether mysqld is locked in memory with --memlock.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log</td>
<td>string</td>
<td>Log connections and queries to file.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_bin</td>
<td>string</td>
<td>Enables binary log.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_bin_trust_function_creators</td>
<td>string</td>
<td>General.Logs</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_bin_trust_routine_creators</td>
<td>string</td>
<td>(deprecated) Use log-bin-trust-function-creators.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_error</td>
<td>string</td>
<td>Error log file.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_output</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_queries_not_using_indexes</td>
<td>string</td>
<td>General.Logs</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_slave_updates</td>
<td>string</td>
<td>General.Logs</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_slow_queries</td>
<td>string</td>
<td>General.Logs</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_update</td>
<td>string</td>
<td>General.Logs</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>log_warnings</td>
<td>long</td>
<td>Log some not critical warnings to the log file.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Name Space Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>long_query_time</td>
<td>double</td>
<td>General.Performance</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>low_priority_updates</td>
<td>string</td>
<td>For non-transactional tables, INSERT, DELETE, and UPDATE statements have lower priority than SELECT statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>lower_case_file_system</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>lower_case_table_names</td>
<td>long</td>
<td>SQL.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_allowed_packet</td>
<td>long</td>
<td>Max packet length to send/receive from to server.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_binlog_cache_size</td>
<td>long</td>
<td>General.Logs</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_binlog_size</td>
<td>long</td>
<td>General.Logs</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_connect_errors</td>
<td>long</td>
<td>Connections.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_connections</td>
<td>long</td>
<td>The number of simultaneous clients allowed.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_delayed_threads</td>
<td>long</td>
<td>General.Performance</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_error_count</td>
<td>long</td>
<td>Max number of errors/warnings to store for a statement.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_heap_table_size</td>
<td>long</td>
<td>Don’t allow creation of heap tables bigger than this.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_insert_delayed_threads</td>
<td>long</td>
<td>This variable is a synonym for max_delayed_threads.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_join_size</td>
<td>long</td>
<td>General.Performance</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_length_for_sort_data</td>
<td>long</td>
<td>Max number of bytes in sorted records.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_prepared_stmt_count</td>
<td>long</td>
<td>Maximum number of prepared</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>---------------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_relay_log_size</td>
<td>long</td>
<td>General.Logs statements in the server.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_seeks_for_key</td>
<td>long</td>
<td>Limit assumed max number of seeks when looking up rows based on a key.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_sort_length</td>
<td>long</td>
<td>The number of bytes to use when sorting BLOB or TEXT values.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_sp_recursion_depth</td>
<td>long</td>
<td>Maximum stored procedure recursion depth.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_tmp_tables</td>
<td>long</td>
<td>Maximum number of temporary tables a client can keep open at a time.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_user_connections</td>
<td>long</td>
<td>Connections.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>max_write_lock_count</td>
<td>long</td>
<td>After this many write locks, allow some read locks to run in between.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>min_examined_row_limit</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>multi_range_count</td>
<td>long</td>
<td>General.Performance</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_data_pointer_size</td>
<td>long</td>
<td>Default pointer size to be used for MyISAM tables.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_max_extra_sort_file</td>
<td>long</td>
<td>Deprecated option.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_max_sort_file_size</td>
<td>long</td>
<td>Table Types.MyISAM</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_mmap_size</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_recover_options</td>
<td>string</td>
<td>The value of the --myisam-recover option.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_repair_threads</td>
<td>long</td>
<td>Table Types.MyISAM</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_sort_buffer_size</td>
<td>long</td>
<td>Memory.Buffers</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_stats_method</td>
<td>string</td>
<td>Specifies how MyISAM index statistics collection code should treat NULLs.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>myisam_use_mmap</td>
<td>string</td>
<td>Indicates whether the server supports connections over named pipes.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>named_pipe</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>net_buffer_length</td>
<td>long</td>
<td>Buffer length for TCP/IP and socket communication.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>net_read_timeout</td>
<td>long</td>
<td>Connections.Networking</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>net_retry_count</td>
<td>long</td>
<td>Connections.Networking</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>net_write_timeout</td>
<td>long</td>
<td>Connections.Networking</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>new</td>
<td>string</td>
<td>Use very new, possibly 'unsafe' functions.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>old</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>old_alter_table</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>old_passwords</td>
<td>string</td>
<td>Use old password encryption method (needed for 4.0 and older clients)</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>open_files_limit</td>
<td>long</td>
<td>General.Directories / Files</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>optimizer_prune_level</td>
<td>long</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>optimizer_search_depth</td>
<td>long</td>
<td>Maximum depth of search performed by the query optimizer.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>optimizer_switch</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>pid_file</td>
<td>string</td>
<td>Pid file used by safe_mysqld.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>plugin_dir</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>port</td>
<td>long</td>
<td>Port number to use for connection.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>preload_buffer_size</td>
<td>long</td>
<td>The size of the buffer that is allocated when preloading indexes.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>prepared_stmt_count</td>
<td>long</td>
<td>The current number of prepared statements.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>profiling</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>profiling_history_size</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>protocol_version</td>
<td>long</td>
<td>The version of the client/server protocol used by the MySQL server.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>pseudo_thread_id</td>
<td>long</td>
<td>Allocation block size for query parsing and execution.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>query_alloc_block_size</td>
<td>long</td>
<td>Allocation block size for query parsing and execution.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>query_cache_limit</td>
<td>long</td>
<td>Don't cache results that are bigger than this.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>query_cache_min_res_unit</td>
<td>long</td>
<td>Memory.Caches.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>query_cache_size</td>
<td>long</td>
<td>The memory allocated to store results from old queries.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>query_cache_type</td>
<td>string</td>
<td>Query cache type.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>query_cache_wlock_invalidate</td>
<td>string</td>
<td>Invalidate queries in query cache on LOCK for write.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>query_prealloc_size</td>
<td>long</td>
<td>Persistent buffer</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-----------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>rand_seed1</td>
<td>long</td>
<td>For query parsing and execution.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>rand_seed2</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>range_alloc_block_size</td>
<td>long</td>
<td>Allocation block size for storing ranges during optimization.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>read_buffer_size</td>
<td>long</td>
<td>Memory.Buffers</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>read_only</td>
<td>string</td>
<td>Make all nontemporary tables read only.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>read_rnd_buffer_size</td>
<td>long</td>
<td>Memory.Buffers</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>relay_log</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>relay_log_index</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>relay_log_info_file</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>relay_log_purge</td>
<td>string</td>
<td>Determines whether relay logs are purged.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>relay_log_space_limit</td>
<td>long</td>
<td>Maximum space to use for all relay logs.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>report_host</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>report_password</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>report_port</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>report_user</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>rpl_recovery_rank</td>
<td>long</td>
<td>Not used.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>secure_auth</td>
<td>string</td>
<td>Disallow authentication for accounts that have old (pre-4.1) passwords.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>secure_file_priv</td>
<td>string</td>
<td>Unknown. Unknown</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>server_id</td>
<td>long</td>
<td>General.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>shared_memory</td>
<td>string</td>
<td>Whether the server allows shared-memory connections.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>shared_memory_base_name</td>
<td>string</td>
<td>The name of shared memory</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------------------------</td>
<td>---------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>skip_external_locking</td>
<td>string</td>
<td>Skip system (external) locking.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>skip_name_resolve</td>
<td>string</td>
<td>Don’t allow SHOW DATABASE commands.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>skip_networking</td>
<td>string</td>
<td>Don’t allow connection with TCP/IP.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>skip_show_database</td>
<td>string</td>
<td>Use compression on master/slave protocol.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slave_compressed_protocol</td>
<td>string</td>
<td>Use compression on master/slave protocol.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slave_exec_mode</td>
<td>string</td>
<td>General.Replication</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slave_load_tmpdir</td>
<td>string</td>
<td>Connections.Networking</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slave_net_timeout</td>
<td>long</td>
<td>Connections.Networking</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slave_skip_errors</td>
<td>string</td>
<td>General.Replication</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slave_transaction_retries</td>
<td>long</td>
<td>General.Replication</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slow_launch_time</td>
<td>long</td>
<td>General.Replication</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slow_query_log</td>
<td>string</td>
<td>Socket file to use for connection.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>slow_query_log_file</td>
<td>string</td>
<td>Socket file to use for connection.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>socket</td>
<td>string</td>
<td>Each thread that needs to do a sort allocates a buffer of this size.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sort_buffer_size</td>
<td>long</td>
<td>Each thread that needs to do a sort allocates a buffer of this size.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_auto_is_null</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_big_selects</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_big_tables</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_buffer_result</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_log_bin</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_log_off</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_log_update</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_low_priority_updates</td>
<td>string</td>
<td>General.Features</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>sql_max_join_size</td>
<td>long</td>
<td>General.Features</td>
</tr>
<tr>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sql_mode</td>
<td>string</td>
<td>Set the SQL server mode.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sql_notes</td>
<td>string</td>
<td>If set to 1, warnings of Note level are recorded.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sql_quote_show_create</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sql_safe_updates</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sql_select_limit</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sql_slave_skip_counter</td>
<td>long</td>
<td>General.Miscellaneous</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sql_warnings</td>
<td>string</td>
<td>CA file in PEM format (check OpenSSL docs, implies --ssl)</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>ssl_ca</td>
<td>string</td>
<td>CA directory (check OpenSSL docs, implies --ssl)</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>ssl_capath</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>ssl_cert</td>
<td>string</td>
<td>X509 certification in PEM format (implies --ssl)</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>ssl_cipher</td>
<td>string</td>
<td>SSL cipher to use (implies --ssl)</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>ssl_key</td>
<td>string</td>
<td>X509 key in PEM format (implies ssl)</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>storage_engine</td>
<td>string</td>
<td>The default storage engine.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sync_binlog</td>
<td>long</td>
<td>Synchronously flush binary log to disk after every #th event.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sync_frm</td>
<td>string</td>
<td>Sync .frm to disk on create. Enabled by default.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sync_replication</td>
<td>long</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sync_replication_slave_id</td>
<td>long</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>mysql variables</td>
<td>sync_replication_timeout</td>
<td>long</td>
<td>Deprecated.</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Name</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>The server system time zone.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>long</td>
<td>Deprecated; use <code>table_open_cache</code> instead.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>long</td>
<td>General.Miscellaneous</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>A synonym for <code>storage_engine</code>.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>long</td>
<td>How many threads to keep in a cache for reuse.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>The stack size for each thread.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>The TIME format (for future)</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>The current time zone.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>General.Miscellaneous</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>long</td>
<td>Memory.Overview</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>Path for temporary files.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>long</td>
<td>Allocation block size for transactions to be stored in binary log.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>long</td>
<td>Persistent buffer for transactions to be stored in binary log.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>The default transaction isolation level.</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>SQL.Overview</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>string</td>
<td>Output version</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>------------------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>version_comment</td>
<td>string</td>
<td>General.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>version_compile_machine</td>
<td>string</td>
<td>The type of machine or architecture on which MySQL was built.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>version_compile_os</td>
<td>string</td>
<td>The type of operating system on which MySQL was built.</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>wait_timeout</td>
<td>long</td>
<td>Connections.Overview</td>
</tr>
<tr>
<td>mysql</td>
<td>variables</td>
<td>warning_count</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>waiting_db</td>
<td>waiting_db</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>waiting_statement</td>
<td>waiting_statement</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>waiting_thread</td>
<td>waiting_thread</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>waiting_time</td>
<td>waiting_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>mysql</td>
<td>waiting_user</td>
<td>waiting_user</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_cache_size</td>
<td>long</td>
<td>The size of the CPU cache.</td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_idle</td>
<td>long_counter</td>
<td>The CPU idle time.</td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_mhz</td>
<td>long</td>
<td>The MHz rating of the CPU.</td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_name</td>
<td>string</td>
<td>The CPU name.</td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_sys</td>
<td>long_counter</td>
<td>CPU system / kernel time.</td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_user</td>
<td>long_counter</td>
<td>CPU user space time.</td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_vendor</td>
<td>string</td>
<td>The name of the CPU vendor.</td>
</tr>
<tr>
<td>os</td>
<td>cpu</td>
<td>cpu_wait</td>
<td>long_counter</td>
<td>CPU IO wait time.</td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_bytes_read</td>
<td>long_counter</td>
<td>The number of bytes read from disk.</td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_bytes_written</td>
<td>long_counter</td>
<td>The number of bytes written to disk.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>---------------</td>
<td>---------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_queue</td>
<td>long</td>
<td>The size of the disk IO queue.</td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_reads</td>
<td>long_counter</td>
<td>The number of disk reads.</td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_rtime</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_servicetime</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_snaptime</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_time</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_writes</td>
<td>long_counter</td>
<td>The number of disk writes.</td>
</tr>
<tr>
<td>os</td>
<td>disk</td>
<td>disk_wtime</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_avail</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_dev_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_dir_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_files</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_flags</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_free_files</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_options</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_sys_type_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_total</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_type_name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_unused</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>fs</td>
<td>fs_used</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>Host</td>
<td>host_id</td>
<td>string</td>
<td>The host unique identifier.</td>
</tr>
<tr>
<td>os</td>
<td>Host</td>
<td>name</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>loadavg</td>
<td>0</td>
<td>double</td>
<td>The total amount of available RAM.</td>
</tr>
<tr>
<td>os</td>
<td>loadavg</td>
<td>1</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>loadavg</td>
<td>2</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>mem</td>
<td>ram_total</td>
<td>long</td>
<td>The total amount of available RAM.</td>
</tr>
<tr>
<td>os</td>
<td>mem</td>
<td>ram_unused</td>
<td>long</td>
<td>The total amount of unused RAM.</td>
</tr>
<tr>
<td>os</td>
<td>mem</td>
<td>swap_page_in</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>mem</td>
<td>swap_page_out</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>os</td>
<td>mem</td>
<td>swap_total</td>
<td>long</td>
<td>The total amount of available swap memory.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>os</td>
<td>mem</td>
<td>swap_unused</td>
<td>long</td>
<td>The total amount of unused swap memory.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>address</td>
<td>string</td>
<td>The address of the network interface.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>broadcast</td>
<td>string</td>
<td>The network interface broadcast address.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>description</td>
<td>string</td>
<td>The network interface description.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>destination</td>
<td>string</td>
<td>The network interface destination address.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>flags</td>
<td>long</td>
<td>The network interface flags.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>hwaddr</td>
<td>string</td>
<td>The network interface MAC address.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>metric</td>
<td>long</td>
<td>Operating System.Net</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>mtu</td>
<td>long</td>
<td>The Maximum Transmission Unit.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>name</td>
<td>string</td>
<td>The network interface name (eth0, eth1 etc.)</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>netmask</td>
<td>string</td>
<td>The network interface subnet mask.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>rx_bytes</td>
<td>long_counter</td>
<td>The number of bytes received.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>rx_dropped</td>
<td>long_counter</td>
<td>The number of received packets dropped.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>rx_errors</td>
<td>long_counter</td>
<td>The number of received packet errors.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>rx_frame</td>
<td>long_counter</td>
<td>The number received packet frame errors.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>rx_overruns</td>
<td>long_counter</td>
<td>The number of received packet overruns.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>rx_packets</td>
<td>long_counter</td>
<td>The number of packets received.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>speed</td>
<td>long</td>
<td>The network interface connection speed.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>tx_bytes</td>
<td>long_counter</td>
<td>The number of bytes transmitted.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>tx_carrier</td>
<td>long_counter</td>
<td>The number of carrier errors on network transmission.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>tx_collisions</td>
<td>long_counter</td>
<td>The number of network transmission collisions.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>tx_dropped</td>
<td>long_counter</td>
<td>The number of network transmissions dropped.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>tx_errors</td>
<td>long_counter</td>
<td>The number of network transmission errors.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>tx_overruns</td>
<td>long_counter</td>
<td>The number of network transmission overruns.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>tx_packets</td>
<td>long_counter</td>
<td>The number of packets transmitted.</td>
</tr>
<tr>
<td>os</td>
<td>net</td>
<td>type</td>
<td>string</td>
<td>The network interface type (ethernet, loopback etc.)</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_arch</td>
<td>string</td>
<td>The architecture of the OS / CPU.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_description</td>
<td>string</td>
<td>Description of the OS.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_machine</td>
<td>string</td>
<td>Machine CPU architecture.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_name</td>
<td>string</td>
<td>The OS name.</td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>--------------------</td>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_patchlevel</td>
<td>string</td>
<td>The OS patch level.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_vendor</td>
<td>string</td>
<td>The OS vendor.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_vendor_code_name</td>
<td>string</td>
<td>The OS vendor code name.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_vendor_name</td>
<td>string</td>
<td>The OS vendor name.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_vendor_version</td>
<td>string</td>
<td>The OS vendor version number.</td>
</tr>
<tr>
<td>os</td>
<td>os</td>
<td>os_version</td>
<td>string</td>
<td>The OS version number.</td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>0</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>1</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>10</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>11</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>12</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>13</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>14</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>15</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>16</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>17</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>18</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>19</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>2</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>20</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>21</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>22</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>23</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>24</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>25</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>26</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>27</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>28</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>29</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>3</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>30</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>31</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>32</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>33</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>Namespace</td>
<td>Namespace Type</td>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>34</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>35</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>36</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>37</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>38</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>39</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>40</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>41</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>42</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>43</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>44</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>45</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>46</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>47</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>48</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>49</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>50</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>51</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>52</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>53</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>54</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>55</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>56</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>57</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>58</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>59</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>60</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>61</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>62</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>7</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>8</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>9</td>
<td>long_counter</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>base</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>util</td>
<td>LogHistogram</td>
<td>nul</td>
<td>long_counter</td>
<td></td>
</tr>
</tbody>
</table>
Appendix H MySQL Enterprise Monitor Release Notes

Release notes for MySQL Enterprise Monitor are published separately. See MySQL Enterprise Monitor 3.0 Release Notes.
MySQL Enterprise Monitor Glossary

These terms are commonly used in information about the MySQL Enterprise Monitor product.

A

advisor
A piece of logic that collects specific data from monitored Assets, optionally saves data to be returned for graphing and reporting purposes, and optionally generates Conditions, which are collected into Events. Advisors can be in one or more Categories.

The MySQL Enterprise Monitor product comes with a number of standard advisors. Based on your organization's particular needs, you may create or adapt custom advisors.
See Also alarm, collection, custom, notification, rule, schedule, standard.

advisor category
A set of Advisors that are designed to enforce MySQL best practices for that specific category. For example, there are predefined advisor categories such as Administration, Availability, Cluster, Replication and Security.
See Also advisor.

Agent
The MySQL Enterprise Monitor component that actively collects data from a host and its MySQL server instances. The data is transmitted to the Service Manager.
See Also instance, Service Manager.

alarm
A term from MySQL Enterprise Monitor 2.x that was replaced with events. Events are handled using event handling, which determines how they act when a potential problem is discovered. For example, an email might be sent.
See Also alert, Critical, event, notification, Warning.

alert
A term from MySQL Enterprise Monitor 2.x that was replaced with events. Events are handled using event handling, which determines how they act when a potential problem is discovered. For example, an email might be sent.
See Also advisor, alarm, event, notification, SMTP, Warning.

Apache
The web server that runs the Tomcat servlet container that powers the UI. The MySQL Enterprise Monitor Service Manager installation includes the Apache web server.
See Also Dashboard, Service Manager, Tomcat.

asset
A component that the MySQL Enterprise Monitor product monitors, such as a MySQL server instance, or a CPU or file system within a server machine. Contrast with metric, which is a property within the component that is measured.
See Also instance, metric.

Asset Selector
The Groups shown in the Asset Selector are the ones you create and manage (on the MySQL Instances Dashboard, or during Agent installation) and those that are discovered automatically from monitoring replication topologies. It is on the left side of many UI pages that controls content in the main-body of the page.
See Also asset, Dashboard, graph.

autodiscovery
The mechanism that detects the set of master, slave, and master/slave servers in a replication configuration, and displays the topology in the Server tree.
See Also Asset Selector, master, replication, slave, topology.
best practice
   Within the MySQL Enterprise Monitor product, a best practice is a specific rule backed up by a metric. When a threshold value is reached, the Service Manager raises an alarm.
   See Also alarm, metric, rule, Service Manager, threshold.

blackout period
   A function of Event Notification, it is a time period where events do not trigger notifications via the Event Handler, in that information is not reported from a MySQL Server. Typically, this is during a maintenance period when the database might go through an unusual workload that does not require raising any notifications.
   See Also Agent, instance, notification.

bottleneck
   An aspect or component of a system whose capacity imposes a limit on performance. In MySQL Enterprise Monitor, you identify bottlenecks in areas such as I/O or memory usage using graphs, and use advisors to automatically raise alerts when problems occur.
   See Also advisor, alert, graph.

bundled
   The “bundled MySQL server” refers to the MySQL server that can be installed along with the MySQL Enterprise Monitor product, to serve as the repository for the collected data.
   See Also repository.

C
   The C programming language.

canonical query
   A synonym for normalized query.
   See Also normalized query.

chain replication
   In replication, a kind of topology where each server acts as a master/slave server, connected to at most two other servers.
   See Also circular replication, master/slave, replication, topology.

circular replication
   In a replication configuration, a topology where every server is both a master and a slave, connected to two other servers, forming a ring structure. Forms a ring structure.
   See Also replication, ring replication, topology.

collection
   A set of related advisors.
   See Also advisor.

Connector
   A software component (analogous to a “driver”) that provide connectivity to the MySQL server for client programs. MySQL comes with connectors for several programming languages and frameworks. The MySQL Enterprise Monitor product includes plugins for certain collectors, to help monitor queries sent by application programs and use that data within the Query Analyzer.
   See Also Connector/J, Connector/Net, Connector/PHP, query, Query Analyzer.

Connector/J
   The Connector for applications written in the Java programming language. It communicates directly with the MySQL server without going through the libmysqlclient shared library.
   See Also Connector, Java.
Connector/Net

The **Connector** for applications written with the .NET programming framework. It communicates directly with the MySQL server without going through the `libmysqlclient` shared library.

See Also .NET, Connector.

Connector/PHP

The **Connector** for applications written in the PHP programming language. The mechanism for communicating with the MySQL server may differ depending on the version of PHP involved.

See Also Connector, PHP.

credentials

The MySQL Enterprise Monitor product interfaces with many different databases and other kinds of servers. Each of these components can have its own login and security credentials. MySQL Enterprise Monitor pulls performance data from the MySQL servers that you monitor, stores the resulting data in a repository that is also a MySQL server, and sends alerts by communicating through other kinds of servers such as SMTP for e-mail alerts and NMS for SNMP traps. It pulls support-related data from the My Oracle Support site. You view the results in the UI, which is protected by its own login and optionally by LDAP authentication.

See Also Dashboard, LDAP, NMS, repository, SMTP, SNMP.

Critical

A high severity level for **alarms**. Within a **rule**, you can designate the **threshold** values that specify whether an alarm triggers a Critical, **Warning**, or **Informational** alert.

See Also advisor, alarm, event, rule, threshold, Warning.

custom

User-created **rules** and **graphs**. Contrast with **standard**.

See Also graph, rule, standard.

D

Dashboard

The Dashboard menu contains several pages (dashboards) for viewing and configuring MySQL Enterprise Monitor. This includes dashboards for the Overview, Replication, and MySQL Instances.

See Also Service Manager, Tomcat, What's New tab.

data collection item

Data values derived from server status variables, operating system status information, and MySQL table information. You can reference these items using mnemonic names in **expressions** when you create or edit **rules**.

See Also expression, rule.

E

event

The highest severity level for an advisor. Within an **Advisor**, you can designate the **threshold** values that specify whether an event triggers an Emergency, **Critical**, **Warning**, or **Informational** event.

See Also Critical, event, Warning.

event

A slightly more comprehensive synonym for **alarm**. Evaluating a **rule** generates an event. The result of an event can be “success” if everything is OK. An alarm is raised only if something goes wrong. You monitor events using the **Events tab**, and by setting **Event Handling**.

See Also alarm, alert, Critical, notification, Warning.

EXPLAIN plan

A textual report showing the internal mechanisms used by a **query**, and estimates for the “cost”, such as amount of data to process, involved in each step. Performance monitoring involves checking whether queries that are slow or frequently run could be sped up or made less resource-intensive, by examining the EXPLAIN plan to check if the MySQL optimizer has chosen the most efficient ways to process indexes, order join
clauses, and so on. The MySQL Enterprise Monitor product includes a number of features for visualizing EXPLAIN plans. These features are language-dependent, relying on support in the various Connectors for programming languages and frameworks. See Also query.

expression
The part of a rule that tests a metric against a threshold value. See Also alarm, metric, rule, threshold.

F

firewall
Additional considerations apply when the server you are monitoring is separated from the Service Manager by a firewall. See Also remote, Service Manager, SSL.

full table scan
An aspect of a SELECT query that often indicates a performance or scalability issue. The query scans every row in a table, rather than using an index to look up a subset of rows. It can be a non-issue for small tables that are cached in memory. It can be unavoidable when querying large tables to prepare reports. Performance issues are most likely when the table being scanned is involved in a join operation, when the query is run frequently, or when the result set only references a small fraction of the rows in the table.

To diagnose possible issues due to full table scans, choose a time period of heavy SQL activity from one of the graphs, use the Query Analyzer to locate queries that process large numbers of rows, and examine the explain plan for the queries. The notation in the explain plan that indicates a full table scan is Type=ALL. See Also EXPLAIN plan, graph, query, Query Analyzer.

G

graph
A visual representation of server activity, resources, or other metrics over time. You monitor graphs using the Graphs tab. See Also Heat Chart, metric.

H

heartbeat
An implementation detail from MySQL Enterprise Monitor 2.x that is no longer present. A heartbeat signal that is sent from an Agent to the Service Manager was defined as the heartbeat, and was present in the connecting URL. See Also Agent, Service Manager.

Heat Chart
A special chart from MySQL Enterprise Monitor 2.3 that showed the status of critical rules. This was replaced with the Overview dashboard. See Also advisor, Dashboard.

host
A computer running one or more instances of MySQL server. Typically each host is monitored by a MySQL Enterprise Monitor Agent. See Also instance, MySQL server.

I

instance
A mysqld daemon running on a MySQL server. There might be multiple instances running on the same server machine. An instance is one of the kinds of asset that the MySQL Enterprise Monitor product can monitor.
The **MySQL Instances** dashboard manages the instances, and the **MySQL Process Discovery** advisor discovers local instances on the host that the Agent is installed on.

See Also **asset**.

**instrument**

A persisted **instance** of a run-time **metric** evaluation. These may store the raw metric data, or the result of an expression or function against a metric. Instruments are generally stored for things that will show in the GUI, such as **graph** data.

See Also **graph**, **metric**.

**inventory**

Core information about a host or MySQL server **instance**. This data is collected by the **Agent**. The inventory includes details such as the MySQL server version number, supported storage engines and **replication** configuration. The data in the inventory helps to determine what other kinds of data can be collected from the hosts and MySQL servers.

See Also **Agent**, **instance**, **replication**.

**J**

**Java**

The MySQL Enterprise **Plugin** for Connector/J enables any application using the Connector/J JDBC driver to supply **query analyzer** information directly to MySQL Enterprise Service Manager. Information is sent to the MySQL Enterprise Service Manager for analysis without any need to modify your applications.

The MySQL Enterprise Monitor product is also partly built on Java technology, using the **Tomcat** servlet container for the web-based GUI. It uses a JVM on the machine that runs the **Service Manager**, its performance is affected by the Tomcat configuration parameters, the Agent is written in Java, and during troubleshooting you might use diagnostic information from the Java environment.

See Also **Connector/J**, **Dashboard**, **JDBC**, **plugin**, **Query Analyzer**, **Tomcat**.

**JDBC**

The underlying database API used by many **Java** applications, including the **Connector/J** component.

See Also **Connector/J**, **Java**.

**L**

**LDAP**

An authentication mechanism that can control access to the **UI**. On Linux, Unix, and OS X systems, you might have a separate LDAP server where each user has their own **credentials**. On Windows systems, the LDAP protocol is used to connect to an Active Directory server for the same authentication purposes. Typically, you map LDAP roles to corresponding MySQL Enterprise Monitor **roles**, to enable groups of users to have basic or administrative access to the Dashboard without configuring each user individually, or giving them a new user ID and password, or requiring extra work to revoke access when they leave the organization.

See Also **credentials**, **Dashboard**, **role**.

**log file**

The MySQL Enterprise Monitor product manages log files for the **Tomcat**, **repository**, and **Service Manager** components, as well as a configuration report pertaining to the initial installation. You can view each of these configuration files through the **UI**, or by examining the physical file.

See Also **Dashboard**, **repository**, **Service Manager**, **Tomcat**.

**Lua**

A programming language that is used for parts of the Aggregator and Service Manager. Although you do not need to know this language to operate the MySQL Enterprise Monitor product, sometimes you specify options containing file paths related to Lua scripts or libraries.

See Also **Agent**, **Query Analyzer**, **Service Manager**.
In a replication configuration, a database server that sends updates to a set of slave servers. It typically dedicates most of its resources to write operations, leaving user queries to the slaves. In complex topologies, a server can be both a master and a slave, known as a master/slave.

See Also master/slave, replication, slave, topology.

In replication, a server that acts as a slave to receive updates from another server, and also acts as a master to propagate changes to another set of slave servers. Keeps the top-level master from having to service too many slaves, and allows certain kinds of changes to be propagated to a subset of slaves. This topology is represented as a tree, with all the intermediate nodes being master/slave servers.

See Also master, replication, slave, topology, tree.

Any property that is measured using a numeric value. Within MySQL Enterprise Monitor, such measurements can be displayed over time as a graph, or an alert can be sent when a threshold value is reached. Each metric is collected from an asset. For example, how much time is taken by a database query, or how full is the file system on a server machine.

See Also alert, asset, graph, threshold.

In a replication configuration, a kind of topology consisting of different combinations of master, slave, and dual-purpose master/slave servers.

See Also chain replication, circular replication, master, master/slave, replication, slave, topology, tree.

To view information about the state, health, activity, and history of a resource such as a MySQL server instance. Monitoring can help to diagnose problems, spot worrisome trends before they turn into problems, reassure when systems are operating normally, and notify when an operator needs to take corrective action. With MySQL Enterprise Monitor, the component you interact with during monitoring is the UI.

See Also Dashboard, instance.

The web site for filing bugs and service requests with Oracle Support. (Commercial customers such as MySQL Enterprise Monitor users now use the official Oracle support channel rather than the MySQL bug database.) In MySQL Enterprise Monitor, you interact with the support site through the What's New tab.

See Also service request, What's New tab.

The MySQL database instance running on a host computer. More than one MySQL instance can run on the same computer, either as separate mysqld processes or through virtualization.

See Also instance.

A programming framework used by many Windows applications, including the Connector/Net component.

See Also Connector/Net.

Acronym for Network Management System, a type of monitoring system that is separate from the MySQL Enterprise Monitor product. If your network has this kind of monitoring capability, MySQL Enterprise Monitor can notify the NMS of network issues by translating events into SNMP traps.

See Also event, SNMP.

A condensed form of the query text used to treat similar queries as if they were identical, for monitoring performance. When MySQL Enterprise Monitor normalizes queries, it disregards differences in keyword
capitalization, whitespace, and most comments. It replaces literal values with placeholders and transforms multi-row insert statements and \texttt{IN} clauses, to group similar statements with different parameters when measuring how much time is consumed by a particular type of query.

See Also canonical query, query.

Notice
A low severity level for events. Within an advisor, you can designate the threshold values that specify the severity level that is triggered.

In 2.x, this was known as the "informational" severity level.

See Also advisor, Critical, event, rule, threshold, Warning.

notification
The e-mail or other form of communication triggered by an alarm. Notifications are defined using Event Handlers.

See Also alarm.

notification group
A collection of users who should receive a notification when advisor alerts occur. These users might have login credentials for the UI, but this is not a requirement.

See Also alert, Dashboard, notification.

P

PHP
A scripting language typically embedded within web pages.

See Also Connector/PHP.

plugin
In the MySQL Enterprise Monitor product, a component that sits between the database and applications that use a particular language-based Connector. It collects query data from both standalone and web-based applications, for MySQL Enterprise Monitor to monitor. For example, there is the MySQL Enterprise Plugin for Connector/Net, and the MySQL Enterprise Plugin for Connector/J. The mechanism for transferring the data, such as whether the data flows through the Aggregator or the Proxy, depends on the particular connector.

See Also Connector, proxy.

proxy
It can accept queries from user applications, forward them to a MySQL server, and return the query results back to the application. You rearrange the TCP/IP ports so that the application communicates with the proxy instead of the database, either by changing the port where the application communicates with the database, or the port where the database listens.

See Also Agent.

Q

QRTi
QRTi stands for "Query Response Time index". It is a "quality of service" measurement for each query, and it uses the Apdex formula for that calculation. For more information about Apdex, see: Apdex on Wikipedia.

See Also Query Analyzer.

QUAN
An abbreviation for Query Analyzer used in some keyword names.

See Also Query Analyzer.

query
In the MySQL Enterprise Monitor context, any SQL statement whose performance might be monitored. Includes not just SQL \texttt{SELECT} queries, but also DML statements such as \texttt{INSERT, UPDATE, and DELETE}, and DDL statements such as \texttt{CREATE TABLE and TRUNCATE TABLE}.
See Also normalized query, slow query log.

Query Analyzer
The MySQL Enterprise Monitor component that tracks data about MySQL queries and summarizes that data using graphs and tables. You interact with it (for example, filtering the displayed queries or selecting a specific time period) using the Query Analyzer tab in the UI.
See Also Dashboard, graph, query.

remote
Additional considerations apply when the server you are monitoring is separated from the Service Manager, for example by a wide-area network or by a firewall.
See Also firewall, remote monitoring, SSL.

remote monitoring
An Agent on one host that monitors different hosts, and MySQL instances on different hosts.
Remote monitoring has limitations, versus local monitoring. Remote monitoring does not provide Operating System level data, such as CPU, file, and network utilization information.
See Also heartbeat, remote.

replication
A set of database features that mirrors the same data across a set of servers. Used for reliability in case of server failure, and to speed up queries by dividing the work across servers. Because replication involves so many aspects of reliability and performance, it is an important aspect to monitor and has a dedicated Replication dashboard in the UI.
See Also Dashboard, Replication dashboard, topology.

Replication dashboard
A dashboard page within the UI that monitors aspects of replication. See Section 14.3, “Replication”.
See Also Dashboard, replication.

repository
The database that stores the monitoring data collected by the MySQL Enterprise Monitor product. It can be a separate database instance that is part of the MySQL Enterprise Monitor installation (the bundled MySQL server), or you can use an existing database of your own. The bundled MySQL server is a level of MySQL database that is fully tested with the MySQL Enterprise Monitor product, and can be kept separate from your other databases to avoid any extra load on them. You might use an existing server that has spare capacity, fast storage devices, tuned configuration parameters, a backup system, or other conveniences that can benefit the MySQL Enterprise Monitor data storage as well.
See Also bundled, instance.

ring replication
See circular replication.

role
A level of access privilege for the UI. One of manager (highest privilege), dba, read-only, or agent (specialized privilege for sending data). Each user account registered with the Dashboard must have one of these roles. Components such as the Agent and the Aggregator use the agent role. To simplify the process of granting credentials for large numbers of users, you can map LDAP roles to these MySQL Enterprise Monitor roles.
See Also Agent, credentials, Dashboard, LDAP.

rule
A test consisting of an expression and one or more threshold values that correspond to different alert levels. When the value of the expression reaches one of the threshold values, an event is generated. Depending on how you configure the rule, the event can result in an alarm, a notification, or both.
See Also advisor, alarm, event, expression, notification, threshold.
schedule
Determined how often each advisor is evaluated.
See Also advisor.

scope
A dynamic filter against an asset class.
See Also asset.

Service Manager
The core MySQL Enterprise Monitor component that receives the monitoring data from the Agent, Aggregator, and Query Analyzer components. It displays this information through the GUI web-interface, and manages the Advisors, Events, and Event Handlers, as well as access to Reporting, Graphing, and Query Analysis.
See Also advisor, Agent, alert, Dashboard, Query Analyzer, rule.

service request
Generic term for a bug that you have filed or a support request that you have opened, on the My Oracle Support site. Formerly, we referred to these as support issues.
See Also My Oracle Support, support issue.

severity
A measure of the seriousness of an Event. Event Handling policies can take into account the severity level in determining the appropriate course of Notification action.
See Also alarm, alert, rule.

slave
In a replication configuration, a database server that receives updates from a master server. Typically used to service user queries, to minimize the query load on the master. In complex topologies, a server can be both a master and a slave, known as a master/slave.
See Also master, master/slave, replication, topology.

slow query log
A MySQL Server facility for tracking queries that consume considerable time and resources. MySQL Enterprise Monitor provides more information about query performance than the slow query log, and does not currently use the slow query log itself.
See Also query.

SMTP
The e-mail protocol for sending alerts as e-mail messages. To send alerts this way, you configure e-mail settings similar to those in an e-mail client.
See Also alert.

SNMP
A protocol for sending event notifications (“SNMP traps”) to an NMS. The MySQL Enterprise Monitor product can turn notifications for selected rules into SNMP traps. In contrast to typical alerts that are only raised when some issue occurs, SNMP traps are broadcast for all state changes, so that corrective action can be cancelled when an issue is cleared.
See Also alert, NMS, notification.

SSL
For MySQL Enterprise Monitor, the default port used for SSL connections is 18443. If you are connecting using SSL, the built-in MySQL Enterprise Monitor certificate is self-signed and may be highlighted as “unsafe” within the browser on initial connection. To prevent problems accessing the site, add an exception for the certificate to your browser for this server.

standard
The predefined rules and graphs supplied as part of the MySQL Enterprise Monitor product. Contrast with custom.
See Also custom, graph, rule.

Status Summary
The Status Summary widget is displayed on top of every page, updates dynamically, and shows current status counters for Hosts monitored, MySQL Instances monitored, MySQL Instances with invalid connection configurations, Unmonitored MySQL Instances, and Emergency Events. The counters are live links for displaying details or resolving the issues they are reporting.

support issue
Former term for bugs and service requests. In the My Oracle Support system, now known as a service request, which is the preferred term.
See Also service request.

T

table scan
See full table scan.

threshold
The cutoff value in a rule. When the rule's expression passes this value, an alert is triggered. There can be different threshold values to trigger the different level alerts.
See Also alert, Critical, expression, rule, Warning.

Tomcat
A component of the Service Manager, the servlet engine that powers the UI.
See Also Apache, Dashboard, Java, Service Manager.

topology
In a replication configuration, the way in which the different master, slave, and dual-purpose master/slave servers are connected. In MySQL Enterprise Monitor, the configurations are classified in the Replication Tab as one of TREE, RING, or MIXED.
See Also master, master/slave, replication, slave.

tree
A data structure often used to represent relationships between MySQL servers. In the UI, servers are displayed in the Server Tree sidebar. In a replication configuration, setting up some machines as dual-purpose master/slave servers produces a nesting relationship that is represented in the Server Tree.
See Also Asset Selector, Dashboard, master/slave, replication.

trending
Predicting when a problem may occur. For example, predicting that a disk will run out of disk space 30 days in the future.

U

UI
The UI is a web-based interface to the MySQL Enterprise Service Manager. The back end is a Java application powered by the Tomcat server.
See Also Dashboard.

UUID
A unique identifier used to distinguish each MySQL instance, host machine, and agent. Because there is so much flexibility in spreading components across multiple systems or running multiple instances and agents on the same system, the combination of these different UUIDs identifies where information came from and the source of any issues. Always generate a new UUID for any one of these components, rather than copying or reusing an existing UUID value.

In MySQL Enterprise Monitor, the UUID for a MySQL server is stored in the table `mysql.inventory`. When a component such as the Agent or a Connector plugin connects to that MySQL server, the applicable MySQL user must have privileges to read this table.
See Also Agent, instance.

W

Warning
A medium severity level for alarms. Within a rule, you can designate the threshold values that specify for each severity level.
See Also advisor, alarm, alert, Critical, event, notification, rule, threshold.

What's New tab
A page within the UI that provides updates and news related to MySQL Enterprise Monitor and your My Oracle Support account.
See Also Dashboard, My Oracle Support.
Index

Symbols

.NET connector plugin, 121

A
Administration Advisors, 187
  32-Bit Binary Running on 64-Bit AMD Or Intel System, 188
  Binary Log Debug Information Disabled, 188
  Binary Logging Is Limited, 188
  Binary Logging Not Enabled, 189
  Binary Logging Not Synchronized To Disk At Each Write, 188
  Binary Logs Automatically Removed Too Quickly, 189
  Database May Not Be Portable Due To Identifier Case Sensitivity, 189
  Event Scheduler Disabled, 189
  General Query Log Enabled, 189
  Host Cache Size Not Sufficient, 190
  In-Memory Temporary Table Size Limited By Maximum Heap Table Size, 190
  InnoDB Status Truncation Detected, 190
  InnoDB Strict Mode Is Off, 191
  InnoDB Tablespace Cannot Automatically Expand, 191
  InnoDB Transaction Logs Not Sized Correctly, 190
  Multiple Threads Used When Repairing MyISAM Tables, 191
  MySQL Server No Longer Eligible For Oracle Premier Support, 191
  Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 192
  No Value Set For MyISAM Recover Options, 192
  Table Cache Set Too Low For Startup, 192
  Time Zone Data Not Loaded, 192
  Warnings Not Being Logged, 192
--adminpassword option, 81
--adminuser option, 80
advisor, 369
advisor category, 369
advisors
  creating, 232, 237
  log file, 156
Advisors, 221
  32-Bit Binary Running on 64-Bit AMD Or Intel System, 188
  Account Has An Overly Broad Host Specifier, 214
  Account Has Global Privileges, 214
  Account Has Old Insecure Password Hash, 215
  Account Has Strong MySQL Privileges, 214
  Account Requires Unavailable Authentication Plugins, 214
  Agent Health Advisor, 221
  Attempted Connections To The Server Have Failed, 194
AUTO_INCREMENT Field Limit Nearly Reached, 209
Average Statement Execution Time Advisor, 228
Binary Log Checksums Disabled, 205
Binary Log Debug Information Disabled, 188
Binary Log File Count Exceeds Specified Limit, 205
Binary Log Row Based Images Excessive, 205
Binary Log Space Exceeds Specified Limit, 205
Binary Log Usage Exceeding Disk Cache Memory Limits, 200
Binary Logging Is Limited, 188
Binary Logging Not Enabled, 189
Binary Logging Not Synchronized To Disk At Each Write, 188
Binary Logs Automatically Removed Too Quickly, 189
Cluster Data Node Data Memory Getting Low, 195
Cluster Data Node Has Been Restarted, 195
Cluster Data Node Index Memory Getting Low, 195
Cluster Data Node Redo Buffer Space Getting Low, 195
Cluster Data Node Redo Log Space Getting Low, 196
Cluster Data Node Undo Buffer Space Getting Low, 196
Cluster Data Node Undo Log Space Getting Low, 196
Cluster Data Nodes Not Running, 196
Cluster DiskPageBuffer Hit Ratio Is Low, 196
Cluster Has Stopped, 196
CPU Utilization Advisor, 225
Database May Not Be Portable Due To Identifier Case Sensitivity, 189
Duplicate MySQL Server UUID, 225
Event Scheduler Disabled, 189
Excessive Disk Temporary Table Usage Detected, 200
Excessive Number of Locked Processes, 200
Excessive Number of Long Running Processes, 200
Excessive Number of Long Running Processes Locked, 200
Excessive Percentage Of Attempted Connections To The Server Have Failed, 194
Filesystem Free Space Advisor, 227
Flush Time Set To Non-Zero Value, 201
General Query Log Enabled, 190
Host Cache Size Not Sufficient, 190
HTTP Server Performance, 198
In-Memory Temporary Table Size Limited By Maximum Heap Table Size, 190
Indexes Not Being Used Efficiently, 201
InnoDB Buffer Cache Has Sub-Optimal Hit Rate, 197
InnoDB Buffer Pool Writes May Be Performance Bottleneck, 201
InnoDB Flush Method May Not Be Optimal, 201
InnoDB Log Buffer Flushed To Disk After Each Transaction, 201

381
InnoDB Log Waits May Be Performance Bottleneck, 202
InnoDB Not Using Newest File Format, 202
InnoDB Status Truncation Detected, 190
InnoDB Strict Mode Is Off, 191
InnoDB Tablespace Cannot Automatically Expand, 191
InnoDB Transaction Logs Not Sized Correctly, 190
Insecure Password Authentication Option Is Enabled, 215
Insecure Password Generation Option Is Enabled, 215
Key Buffer Size May Not Be Optimal For Key Cache, 197
LOCAL Option Of LOAD DATA Statement Is Enabled, 215
Master Not Verifying Checksums When Reading From Binary Log, 206
Maximum Connection Limit Nearing Or Reached, 194
Multiple Threads Used When Repairing MyISAM Tables, 191
MyISAM Concurrent Insert Setting May Not Be Optimal, 202
MyISAM Indexes Found with No Statistics, 211
MySQL Agent Memory Usage Excessive, 193
MySQL Agent Not Reachable, 193
MySQL Availability, 193
MySQL Enterprise Backup Health Advisor, 223
MySQL Process Discovery Advisor, 224
MySQL Server Has Been Restarted, 194
MySQL Server No Longer Eligible For Oracle Premier Support, 191
Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 192
No Value Set For MyISAM Recover Options, 192
Non-root User Has DB, Table, Or Index Privileges On All Databases, 216
Non-root User Has GRANT Privileges On All Databases, 216
Non-root User Has Server Admin Privileges, 216
Object Changed: Database Has Been Altered, 210
Object Changed: Database Has Been Created, 210
Object Changed: Database Has Been Dropped, 210
Object Changed: Function Has Been Created, 210
Object Changed: Function Has Been Dropped, 210
Object Changed: Index Has Been Created, 211
Object Changed: Index Has Been Dropped, 211
Object Changed: Table Has Been Altered, 212
Object Changed: Table Has Been Created, 212
Object Changed: Table Has Been Dropped, 212
Object Changed: User Has Been Dropped, 213
Object Changes Detected, 211
Policy-Based Password Validation Does Not Perform Dictionary Checks, 216
Policy-Based Password Validation Is Weak, 216
Policy-Based Password Validation Not Enabled, 217
Prepared Statements Not Being Used Effectively, 203
Prepared Statements Not Being Used Effectively, 203
Privilege Alterations Detected: Privileges Granted, 217
Privilege Alterations Detected: Privileges Revoked, 217
Privilege Alterations Have Been Detected, 217
Query Analysis Reporting, 229
Query Cache Has Sub-Optimal Hit Rate, 197
Query Cache Is Excessively Fragmented, 203
Query Cache Potentially Undersized, 197
Query Pileup Advisor, 229
Replication Configuration Advisor, 206
Replication Status Advisor, 206
Root Account Can Login Remotely, 217
Root Account Without Password, 217
Server Contains Default "test" Database, 218
Server Has Accounts Without A Password, 218
Server Has Anonymous Accounts, 218
Server Has No Locally Authenticated Root User, 218
Server Includes A Root User Account, 218
Server-Enforced Data Integrity Checking Disabled, 211
Server-Enforced Data Integrity Checking Not Strict, 212
SHA-256 Password Authentication Not Enabled, 219
Slave Detection Of Network Outages Too High, 206
Slave Execution Position Too Far Behind Read Position, 207
Slave Has Login Accounts With Inappropriate Privileges, 207
Slave Master Info/Relay Log Info Not Crash Safe, 207
Slave Not Configured As Read Only, 207
Slave Not Verifying Checksums When Reading From Relay Log, 207
Slave Relay Log Space Is Very Large, 208, 208
Slave SQL Processing Not Multi-Threaded, 208
Slave SQL Thread Reading From Older Relay Log Than I/O Thread, 208
Slave Too Far Behind Master, 208
Slave Without REPLICATION SLAVE Accounts, 209
SQL Statement Generates Warnings or Errors, 229
Support Diagnostics, 199
Symlinks Are Enabled, 219
Table Cache Not Optimal, 197
Table Cache Set Too Low For Startup, 192
Table Lock Contention Excessive, 203
Tables Found with No Primary or Unique Keys, 212
Thread Cache Not Enabled, 203
Thread Cache Size May Not Be Optimal, 198
Thread Pool Stall Limit Too Low, 203
Thread Pooling Not Enabled, 204
Time Zone Data Not Loaded, 192
Too Many Concurrent Queries Running, 204
User Has Rights To Database That Does Not Exist, 219
User Has Rights To Table That Does Not Exist, 219
Users Can View All Databases On MySQL Server, 219
Warnings Not Being Logged, 192
Wrong Version Agent Tracker, 198
agent
installation, 33
log file, 156
troubleshooting, 58
Agent, 369
Agent Advisors, 193
Agent Health Advisor, 221
MySQL Agent Memory Usage Excessive, 193
MySQL Agent Not Reachable, 193
agent.sh
agent.bat, 287
--agent_autocreate option, 87
--agent_installtype option, 88
--agentpassword option, 83
--agentservicename option, 89
--agentuser option, 84
Aggregator
PHP connector, 113
alarm, 369
alert, 369
Apache, 369
log file, 156
Apple OS X, 45, 54, 98, 281
architecture, 5
asset, 369
Asset Selector, 130, 369
autodiscovery, 369
Availability Advisors, 193
Attempted Connections To The Server Have Failed, 194
Excessive Percentage Of Attempted Connections To
The Server Have Failed, 194
Maximum Connection Limit Nearing Or Reached, 194
MySQL Availability, 193
MySQL Server Has Been Restarted, 194
B
backup
restore, 69
Backup Advisors
MySQL Enterprise Backup Health Advisor, 223
--backupdir option, 77
backups, 278
best practice, 370
blackout period, 244, 370
bottleneck, 370
browsers, 277
bundled, 370
C
C, 370
canonical query, 370
Catalina
log file, 156
chain replication, 370
ChangeLog, 367
changes
release notes, 367
--checkmysqlohost option, 84
circular replication, 370
Cluster Advisors, 195
Cluster Data Node Data Memory Getting Low, 195
Cluster Data Node Has Been Restarted, 195
Cluster Data Node Index Memory Getting Low, 195
Cluster Data Node Redo Buffer Space Getting Low, 195
Cluster Data Node Redo Log Space Getting Low, 196
Cluster Data Node Undo Buffer Space Getting Low, 196
Cluster Data Node Undo Log Space Getting Low, 196
Cluster Data Nodes Not Running, 196
Cluster DiskPageBuffer Hit Ratio Is Low, 196
Cluster Has Stopped, 196
collection, 370
config.properties file, 282
config.sh
config.bat, 286
configuration files, 281
Connector, 370
Connector/J, 117, 370
Connector/Net, 121, 371
Connector/PHP, 371
--createBackup option, 83
--createDataBackup option, 77
credentials, 371
Critical, 371
custom, 371
custom data collection
customizing advisors, 240
D
Dashboard, 371
data collection item, 371
data collection items, 301
data_collection_interval option, 283
--dbhost option, 82
--dbname option, 82
dbpool.default.initialSize option, 285
dbpool.default.maxActive option, 285
dbpool.default.maxIdle option, 285
dbpool.default.maxWaitMillis option, 285
dbpool.default.minEvictableIdleTimeMillis option, 285
dbpool.default.minIdle option, 285
dbpool.default.timeBetweenEvictionRunsMillis option, 285
dbpool.ui.initialSize option, 284
dbpool.ui.maxActive option, 284
dbpool.ui.maxIdle option, 284
dbpool.ui.maxWaitMillis option, 284
dbpool.ui.minEvictableIdleTimeMillis option, 285
dbpool.ui.minIdle option, 284
dbpool.ui.timeBetweenEvictionRunsMillis option, 285
--dbport option, 81
--debuglevel option, 78, 84
--debugtrace option, 78, 84
--dbport option, 81
--debuglevel option, 78, 84
--debugtrace option, 78, 84
Diagnostics Report, 159

E
event, 371, 371
Events, 165
EXPLAIN plan, 371
expression, 372
expressions, 235

F
FAQs, 269
firewall, 372
firewall issues, 57
--forceRestart option, 82
full table scan, 372

G
--generalpassword option, 87
--generaluser option, 87
Global Settings, 149
graph, 372
graphs
  creating, 232, 239

H
heartbeat, 372
Heat Chart, 372
--help option, 77, 89
Hibernate
  log file, 156
host, 372

I
--ignore-old-proxy-aggr option, 88
InnoDB Buffer Pool Usage Report, 162
installation
  agent, 33
  backup, 69
  post-install tasks, 61
  service manager, 25
  unattended, 75
  uninstalling, 95
  upgrading, 67
--installdir option, 79, 84
--installer option, 79
--installer-language option, 85
instance, 372
instrument, 373
internal_perf_enable option, 283
internal_perf_server_id option, 283
inventory, 373

J
Java, 373
Java connector plugin, 117
Java VM
  installation, 25
JDBC, 373
  log file, 156

L
LDAP, 373
limitations, 277
--limitedpassword option, 87
--limiteduser option, 87
Linux, 37, 55, 96, 281
locale, 32
log file, 373
  Monitor UI for all components, 156
MySQL Enterprise Service Manager, 281
Lua, 373

M
Mac OS X, 45, 54, 98, 281
Manage Users, 155
--managerhost option, 85
--managerport option, 85
master, 374
master/slave, 374
Memory Usage Advisors, 196
  InnoDB Buffer Cache Has Sub-Optimal Hit Rate, 197
  Key Buffer Size May Not Be Optimal For Key Cache, 197
  Query Cache Has Sub-Optimal Hit Rate, 197
  Query Cache Potentially Undersized, 197
  Table Cache Not Optimal, 197
  Thread Cache Size May Not Be Optimal, 198
metric, 374
MIB file, 281
mixed replication, 374
--mode option, 78, 85
monitor, 374
  installation, 25
Monitoring and Support Services Advisors, 198
  Duplicate MySQL Server UUID, 225
  HTTP Server Performance, 198
  MySQL Process Discovery Advisor, 224
  Support Diagnostics, 199
  Wrong Version Agent Tracker, 198
My Oracle Support, 374
MySQL Enterprise Backup, 278
MySQL Server
  installation, 25
MySQL server, 374
mysql-monitor-agent.log file, 286
  --mysqlconnectiongroup option, 89
  --mysqlconnmethod option, 86
mysqldump, 278
mysqldump, 278
mysqlenterprise.* options, 113, 117
--mysqlhost option, 86
--mysql-identity-source option, 80
--mysqlpassword option, 86
--mysqlport option, 86
--mysqlsocket option, 86
--mysqluser option, 87
--mysql_installation_type option, 81
--mysql_ssl option, 80

N
.NET, 374
NMS, 374
normalized query, 374
Notice, 375
notification, 375
notification group, 375
notify_thread_pool_size option, 283

O
Operating System Advisors, 199
  CPU Utilization Advisor, 225
  Filesystem Free Space Advisor, 227
--optionfile option, 78, 87
OS X, 45, 54, 98, 281
overview, 5
Overview Dashboard, 135

P
Performance Advisors, 199
  Binary Log Usage Exceeding Disk Cache Memory Limits, 200
  Excessive Disk Temporary Table Usage Detected, 200
  Excessive Number of Locked Processes, 200
  Excessive Number of Long Running Processes, 200
  Excessive Number of Long Running Processes Locked, 200
  Flush Time Set To Non-Zero Value, 201
  Indexes Not Being Used Efficiently, 201
  InnoDB Buffer Pool Writes May Be Performance Bottleneck, 201
  InnoDB Flush Method May Not Be Optimal, 201
  InnoDB Log Buffer Flushed To Disk After Each Transaction, 201
  InnoDB Log Waits May Be Performance Bottleneck, 202
  InnoDB Not Using Newest File Format, 202
  MyISAM Concurrent Insert Setting May Not Be Optimal, 202
  Prepared Statements Not Being Closed, 202
  Prepared Statements Not Being Used Effectively, 203
  Query Cache Is Excessively Fragmented, 203
  Table Lock Contention Excessive, 203
  Thread Cache Not Enabled, 203

Thread Pool Stall Limit Too Low, 203
Thread Pooling Not Enabled, 204
Too Many Concurrent Queries Running, 204
performance schema, 252
performance tuning, 91
PHP, 375
PHP connector plugin
  Aggregator, 113
ports
  LDAP, 152
  MySQL server for monitoring, 86
  MySQL server for repository, 81
  service manager, 79, 85
  Tomcat, 79
  Tomcat SSL, 80
post-install tasks, 61
proxy, 375

Q
QRTI, 375
QUAN, 375
quanal.collect option, 284
query, 375
Query Analysis Advisors, 228
  Average Statement Execution Time Advisor, 228
  Query Analysis Reporting, 229
  Query Pileup Advisor, 229
  SQL Statement Generates Warnings or Errors, 229
Query Analyzer, 132, 251, 376
  .NET connector, 121
  Java connector, 117
  PHP connector, 113
  supplying query data, 251
Query Analyzer tab, 256

R
release notes, 367
remote, 376
remote monitoring, 376
Replication, 138
replication, 376
log file, 156
Replication Advisors, 204
  Binary Log Checksums Disabled, 205
  Binary Log File Count Exceeds Specified Limit, 205
  Binary Log Row Based Images Excessive, 205
  Binary Log Space Exceeds Specified Limit, 205
  Master Not Verifying Checksums When Reading From Binary Log, 206
  Replication Configuration Advisor, 206
  Replication Status Advisor, 206
  Slave Detection Of Network Outages Too High, 206
  Slave Execution Position Too Far Behind Read Position, 206
  Slave Has Login Accounts With Inappropriate Privileges, 207
Slave Master Info/Relay Log Info Not Crash Safe, 207
Slave Not Configured As Read Only, 207
Slave Not Verifying Checksums When Reading From Relay Log, 207
Slave Relay Log Space Is Very Large, 207, 208
Slave SQL Processing Not Multi-Threaded, 208
Slave SQL Thread Reading From Older Relay Log Than I/O Thread, 208
Slave Too Far Behind Master, 208
Slave Without REPLICATION SLAVE Accounts, 209
Replication dashboard, 376
--restartImmediately option, 83
repository, 376
--restore Immediately option, 83
restore
backup, 69
restoring
backup, 69
rule, 376
rules
blackout periods, 244
creating, 232
variable substitution, 235

Schedule, 377
Schema Advisors, 209
AUTO_INCREMENT Field Limit Nearly Reached, 209
MyISAM Indexes Found with No Statistics, 211
Object Changed: Database Has Been Altered, 210
Object Changed: Database Has Been Created, 210
Object Changed: Database Has Been Dropped, 210
Object Changed: Function Has Been Created, 210
Object Changed: Function Has Been Dropped, 210
Object Changed: Index Has Been Created, 211
Object Changed: Index Has Been Dropped, 211
Object Changed: Table Has Been Altered, 212
Object Changed: Table Has Been Created, 212
Object Changed: Table Has Been Dropped, 212
Object Changed: User Has Been Dropped, 213
Object Changes Detected, 211
Server-Enforced Data Integrity Checking Disabled, 211
Server-Enforced Data Integrity Checking Not Strict, 211
Tables Found with No Primary or Unique Keys, 212
Scope, 377
Security Advisors, 213
Account Has An Overly Broad Host Specifier, 214
Account Has Global Privileges, 214
Account Has Old Insecure Password Hash, 215
Account Has Strong MySQL Privileges, 214
Account Requires Unavailable Authentication Plugins, 214
Insecure Password Authentication Option Is Enabled, 215
Insecure Password Generation Option Is Enabled, 215
LOCAL Option Of LOAD DATA Statement Is Enabled, 215
Non-root User Has DB, Table, Or Index Privileges On All Databases, 216
Non-root User Has GRANT Privileges On All Databases, 216
Non-root User Has Server Admin Privileges, 216
Policy-Based Password Validation Does Not Perform Dictionary Checks, 216
Policy-Based Password Validation Is Weak, 216
Policy-Based Password Validation Not Enabled, 217
Privilege Alterations Detected: Privileges Granted, 217
Privilege Alterations Detected: Privileges Revoked, 217
Privilege Alterations Detected: Privileges Have Been Detected, 217
Root Account Can Login Remotely, 217
Root Account Without Password, 217
Server Contains Default "test" Database, 218
Server Has Accounts Without A Password, 218
Server Has Anonymous Accounts, 218
Server Has No Locally Authenticated Root User, 218
Server Includes A Root User Account, 218
SHA-256 Password Authentication Not Enabled, 219
Symlinks Are Enabled, 219
User Has Rights To Database That Does Not Exist, 219
User Has Rights To Table That Does Not Exist, 219
Users Can View All Databases On MySQL Server, 219
service manager
database name, 82
installation, 25
Service Manager, 377
service request, 377
services
starting and stopping, 29
severity, 377
slave, 377
slow query log, 377
SMTP, 377
SNMP, 377
SNMP traps, 281
SQL
log file, 156
SSH tunnelling, 57
SSL, 80, 377
standard, 377
starting
MySQL Enterprise Monitor service, 30
MySQL Enterprise Monitor services, 29
Status Summary, 129, 378
stopping
MySQL Enterprise Monitor service, 30
MySQL Enterprise Monitor services, 29
support issue, 378
supportReport.retention.minutes option, 284
--system-size option, 79

T
thread_pool_size option, 283
threshold, 378
thresholds, 235
Time Series Graphs, 161
timezone, 32
Tomcat, 378
    installation, 25
    log file, 156
    starting and stopping, 30
    --tomcatport option, 79
    --tomcatsslport option, 80
topology, 378
tree, 378
trending, 378
troubleshooting, 58
tuning, 91

U
UI, 378
ui.javascript.useClientSideStorage option, 284
unattended installation, 75
--unattendedmodeui option, 81, 88
uninstalling, 95
Unix, 37, 55, 96, 281
upgrading, 67
--use-external-glib option, 88
User Preferences, 158
UUID, 378

V
variable substitution, 235
--version option, 77, 88

W
Warning, 379
Web browsers, 278
What's New, 132
What's New tab, 379
Wiki markup, 236
Windows, 30, 42, 53, 95, 281