

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
III Using MySQL Enterprise Monitor	125
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
21	GUI-Based Advisor Reference	221
21.1	Agent Health Advisor	221
21.2	MySQL Enterprise Backup Health Advisor	223
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
22	Customizing MySQL Enterprise Monitor	231
22.1	Customizing Groups	231
22.2	Creating Advisors and Rules	232
22.2.1	Creating Advisors	232
22.2.2	Overview of Graph Creation	233
22.2.3	Overview of Advisor Creation	234
22.2.4	Variables	235
22.2.5	Thresholds	235
22.2.6	Using Strings	236
22.2.7	Wiki Format	236
22.2.8	Creating a New Advisor: An Example	237
22.2.9	Creating a New Graph: An Example	239
22.3	Custom Data Collection	240
22.3.1	Custom.xml	240
22.3.2	Queries	241
22.3.3	Data Collection Attributes	242
22.4	Event Notification Blackout Periods	244
22.4.1	Scripting Blackouts	245
IV	Using the Query Analyzer	247
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
V Appendices	265
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 <code>config.properties</code> 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
MySQL Enterprise Monitor Glossary	369
Index	381

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 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

21.6 Filesystem - General	227
21.7 Filesystem - Estimated Full Capacity	227
21.8 Filesystem - Percentage of Space	228
21.9 Filesystem - Percentage Used in Time Range	228
22.1 Manage Instances example	232
22.2 Creating a New Advisor	238
23.1 MySQL Enterprise Monitor User Interface: Query Analyzer	256
23.2 MySQL Enterprise Monitor User Interface: Canonical Query Tab for a Query	258
23.3 MySQL Enterprise Monitor User Interface: Example Query Tab for a Query	259
23.4 MySQL Enterprise Monitor User Interface: Explain Query Tab for a Query	260
23.5 Graphs Tab for a Query	260
23.6 MySQL Enterprise Monitor User Interface: Correlated Graphs	261
23.7 MySQL Enterprise Monitor User Interface: Query Analyzer Filter Options	262
23.8 MySQL Enterprise Monitor User Interface: Query Analyzer Configuration	263

List of Tables

3.1 Disk space Required	20
4.1 Installation Parameters	26
6.1 Default Root Path (<i>\$INSTALL_ROOT</i>)	62
6.2 SSL Configuration Options For The Agent's <i>bootstrap.properties</i>	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 <i>config.properties</i> File	282
D.4 Optional <i>config.properties</i> 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/>.

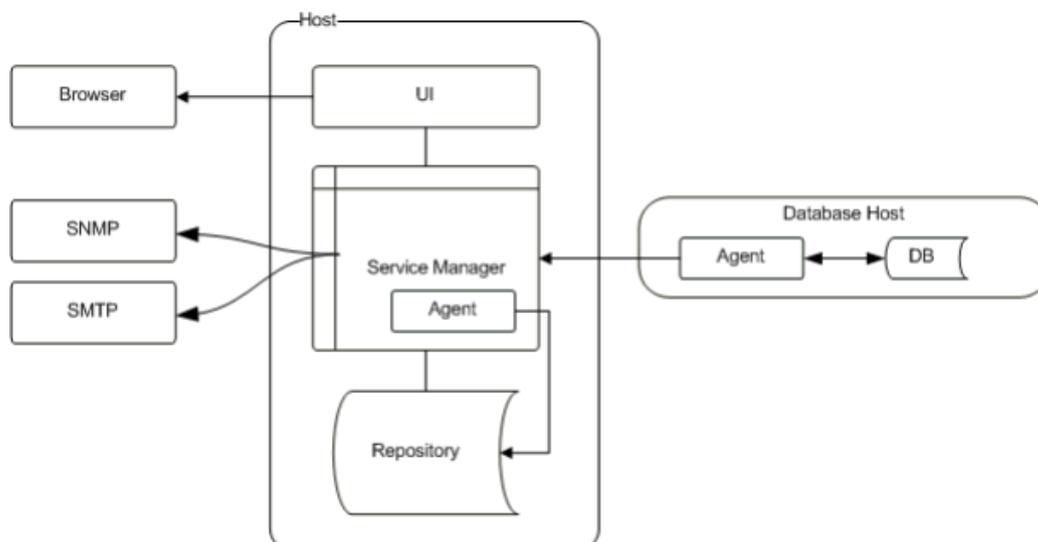
MySQL Enterprise Monitor is a companion product to MySQL Server that enables monitoring of MySQL instances and their hosts, notification of potential issues and problems, and advice on how to correct issues. MySQL Enterprise Monitor can monitor all types of installation, from a single MySQL instance 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

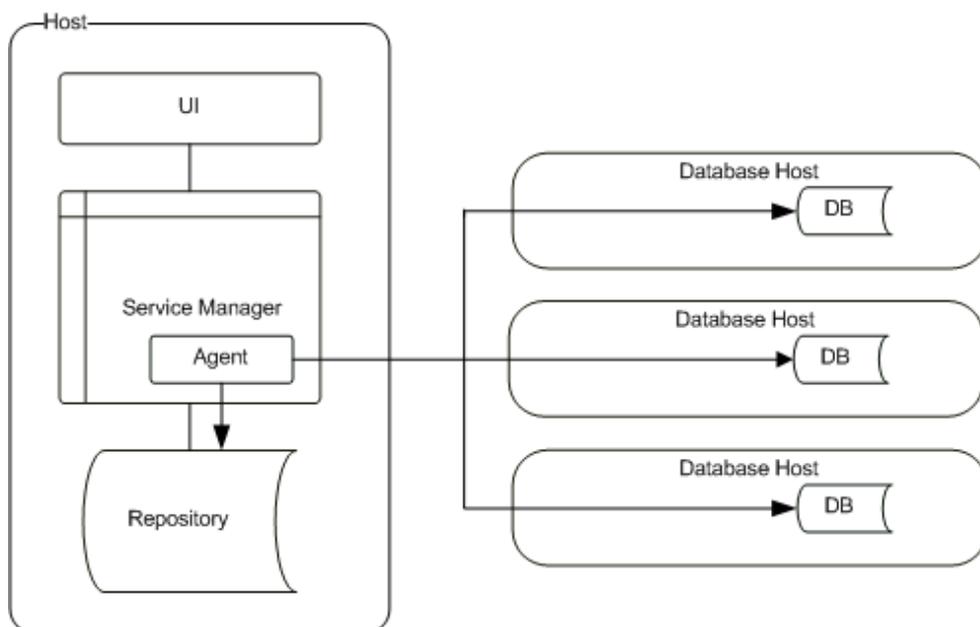


MySQL Enterprise Monitor has the following components:

- 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



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.

- 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:

- 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.

- **Schema:** Identifies schema changes.
- **Security:** Checks MySQL Servers for known security issues.

It is also possible to create custom Advisors.

The 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 Agent to stop collecting that data.

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?



Note

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

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.



Note

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.

- 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.

[2] Known limitation: Process discovery is not supported on Microsoft Windows.

[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

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

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-platform-installer.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 3.1 Disk space Required

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

**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.

**Important**

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.

**Note**

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.

**Important**

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.

**Important**

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.

**Note**

It is not possible to use MySQL 5.1.x, or 5.5.x, for the MySQL Enterprise Monitor repository.

- 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”](#).

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

Parameter	Small	Medium	Large
Tomcat Heap Size	256MB	768MB	2048MB
Tomcat MaxPermSize	200MB	512MB	1024MB
table_definition_cache	800	2048	2048
innodb_buffer_pool_size	100MB	768MB	8096MB



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:

**Note**

During the installation process you must enter usernames and passwords for components of the Enterprise Monitor. Make note of these in a secure location so you can recover them in case they are forgotten.

3. Click **OK** to continue.
4. On the **Welcome** dialog, click **Forward**.
The **Installation Directory** dialog is displayed.
5. 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>./mysqlmonitor-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)
./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:

```
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:

```
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.

- 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.

- **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.threads` 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.threads` 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.threads` 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.

- **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 that 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
    Linux Standard Base 3.2 init script function
ii  lsb-release                 3.2-20ubuntu4
```

```
Linux Standard Base version reporting utility
```

Alternatively, you can use the `lsb_release` command. Existence of this command normally indicates that the current distribution is LSB compliant.

To install the agent, navigate to the directory that contains the file `mysqlmonitoragent-version-installer.bin` (where `version` indicates the three-part version number, the OS, and the architecture).



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:

```
Language Selection

Please select the installation language
[1] English - English
[2] Japanese - 日本語
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]:
```

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

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

- 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

Hostname or IP address []: 127.0.0.1

Tomcat SSL Port [18443]:

Agent Username [agent]:

Agent Password :
Re-enter :
```

- 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:

```
-----
Monitored Database Information

IMPORTANT: The Admin user account specified below requires special MySQL
privileges. Please see the User Guide for more information.

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”](#).

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.

- 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]:
```

- 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 :
```

- 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
```

- 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
```

```

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

Info to start the 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

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



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".

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**

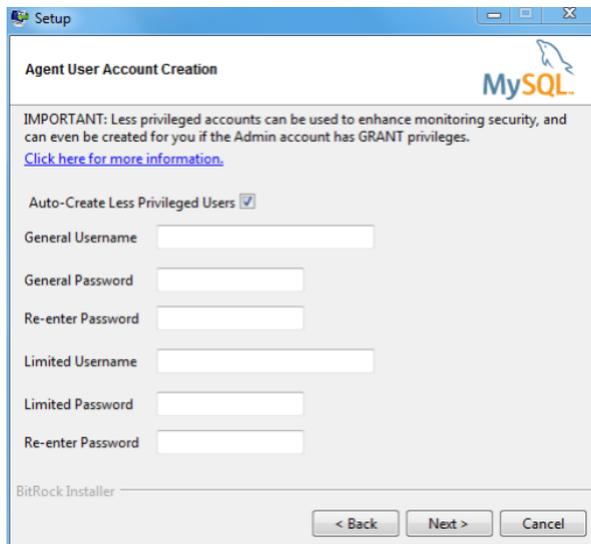
Currently, on Windows, the monitor agent only includes support for connecting to the server to be monitored via TCP/IP, so you cannot monitor a server running with the `--skip-networking`.

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.

- 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



- 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.
- You have a final opportunity to change the installation parameters. Click **Next** to start the installation process.
- Once the agent is installed, you get a confirmation message. Click **Next** to finalize the installation.
- 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:

- 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:

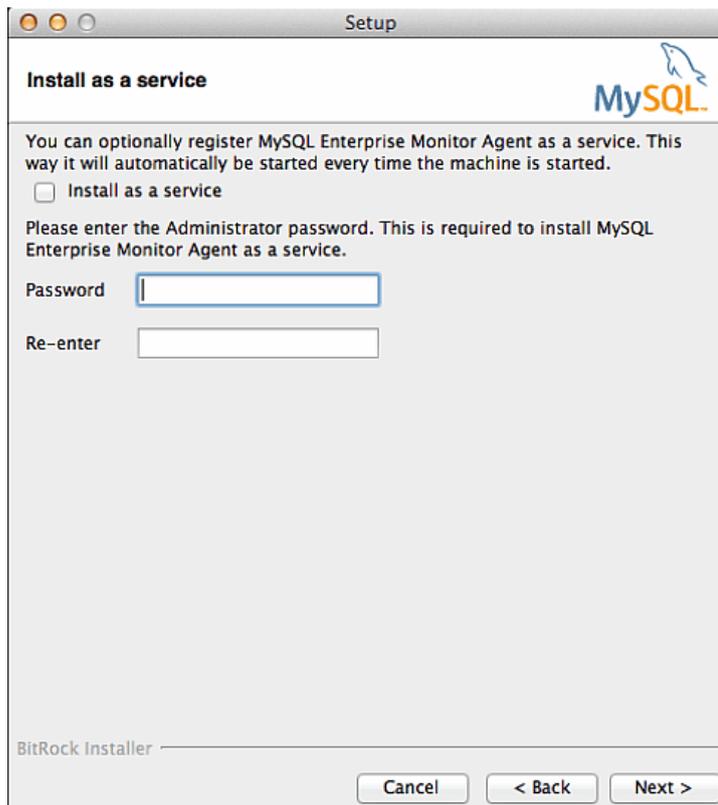
Figure 5.5 MySQL Enterprise Monitor: Agent Monitoring Options



- **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.

Figure 5.6 MySQL Enterprise Monitor: Install as a service

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".

Figure 5.7 MySQL Enterprise Monitor: Installing Agent on Mac OS X: MySQL Enterprise Service Manager Options

Setup

MySQL Enterprise Monitor Options

MySQL

Hostname or IP address

Tomcat SSL Port

The following are the username and password that the Agent will use to connect to the Monitor. They were defined when you installed the Monitor. They can be modified under Settings, Manage Users. Their role is defined as "agent".

Agent Username

Agent Password

Re-enter

BitRock Installer

Cancel < Back Next >

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.

Figure 5.8 MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information (TCIP)

Setup

Monitored Database Information 

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

MySQL hostname or IP address

Validate MySQL hostname or IP address

MySQL Port

Admin User

Admin Password

Re-enter Password

Monitor Group

BitRock Installer

Cancel < Back Next >

- 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`.

Figure 5.9 MySQL Enterprise Monitor: Installing Agent on Mac OS X: Monitored Database Information (Socket)

Setup

Monitored Database Information 

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

MySQL Socket

Admin User

Admin Password

Re-enter Password

Monitor Group

BitRock Installer

Cancel < Back Next >

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"](#).

Figure 5.10 MySQL Enterprise Monitor: Installing Agent on OS X: Additional Agent Users

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.

**Warning**

If you are using Query Analyzer, turning off the agent prevents your applications from communicating with the MySQL server. See [Section 13.2, “The Query Analyzer”](#).

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:

```
shell> net stop MySQLEnterpriseMonitorAgent
```

**Note**

`MySQLEnterpriseMonitorAgent` is the default name of the Monitor Agent service.

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.

- For replication autodiscovery, do **not** use remote monitoring with replication slaves or masters. The Agent must be installed on the same machine as the server you are monitoring in order for discovery to work properly. For more information, see [Section 14.3, “Replication”](#).

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:

```
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.

- 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 6.1 Default Root Path (`$INSTALL_ROOT`)

Operating System	Default Path
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\</code>
Linux / Solaris	<code>/opt/mysql/enterprise/monitor/</code>
Mac OS X	<code>/Applications/mysql/enterprise/monitor/</code>

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.

**Important**

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.

**Important**

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`

Parameter	Values	Description	Removed
<code>ssl-verify-hostnames</code>	<code>true</code> or <code>false</code>	Verify that the hostname of the service manager that the Agent is connected to matches what is in the SSL certificate, Default is <code>false</code> , as we are only using SSL for confidentiality	
<code>ssl-allow-self-signed-certs</code>	<code>true</code> or <code>false</code>	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> .	
<code>ssl-verify-host-certs</code>	<code>true</code> or <code>false</code>	Default <code>false</code> , but to support self-signed certificates, a commercial certificate, or if the CA certificate has been imported into a keystore, then set to <code>true</code> .	3.0.20
<code>ssl-ca-keystore-path</code>	String	Path to keystore with CA cert(s), if <code>ssl-allow-self-signed-certs</code> is <code>true</code> . This path must be defined as a URL. For example: <code>file:///Applications/mysql/enterprise/agent/etc/mykeystore</code>	
<code>ssl-ca-keystore-password</code>	String	Password for the CA keystore, if <code>ssl-allow-self-signed-certs</code> is <code>true</code> .	

An example `bootstrap.properties` SSL certification section:

```
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
```

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, ST=Uppsala, C=SE
Issuer: O=MySQL AB, L=Uppsala, ST=Uppsala, C=SE
Serial number: 100002
Valid from: Fri Jan 29 12:56:49 CET 2010 until: Wed Jan 28 12:56:49 CET 2015
Certificate fingerprints:
  MD5:  E5:FB:56:76:78:B1:0C:D7:B0:80:9F:65:06:3E:48:3E
  SHA1: 87:59:80:28:CE:15:EF:7E:F1:75:4B:76:77:5E:64:EA:B7:1D:D1:18
  SHA256: F4:0B:79:52:CF:F3:A1:A4:7F:B2:D7:C1:65:60:F0:80:93:87:D2:68:9A:A1:
         84:F4:06:6E:8E:CF:C1:F6:1B:52
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
```

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:

```
5a:86:16:fb:2e:16:e8:21:ef:07:ee:6c:fc:4f:84:e5
```

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:

```
agent-host-id=ssh:{5a:86:16:fb:2e:16:e8:21:ef:07:ee:6c:fc:4f:84:e5}
```

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.

```
$ 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:

```
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 [mysqlenterprise/monitor/backup/java/lib/security/cacerts](#) 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 begins 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

to do so safely. You can then leave the 2.3 Monitor running indefinitely to view historical graph and Events data, or you can uninstall it and reclaim the disk space. Make sure to disable notifications if you intend to continue using it to view past monitoring data.

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.

AUpgrading 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
--mysqlhost 127.0.0.1 --mysqlport 3306
--mysqluser root --mysqlpassword foo --agent_autocreate
--limiteduser limited --limitedpassword foo --generaluser general --generalpassword foo
--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`:

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

```
mysql-identity-source=host_plus_datadir
```

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
```

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 8.1 MySQL Enterprise Service Manager Installer Options

Format	Description	Removed
<code>--adminpassword</code>	Password for the database repository	
<code>--adminuser</code>	Username for the database repository	
<code>--backupdir</code>	Backup directory path.	
<code>--createDataBackup</code>	Backup stored data. Upgrade process only.	

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

- `--help`

Command-Line Format	<code>--help</code>
----------------------------	---------------------

Display the list of valid installer options.

- `--version`

Command-Line Format	<code>--version</code>
----------------------------	------------------------

Display product and version information.

- `--backupdir`

Command-Line Format	<code>--backupdir</code>	
Permitted Values	Type	string

Upgrade only. The backup directory.

- `--createDataBackup`

Command-Line Format	<code>--createDataBackup</code>	
Permitted Values	Type	boolean
	Default	1
	Valid Values	0 (Do not create data backup)

	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	<code>--optionfile</code>
----------------------------	---------------------------

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

- `--mode`

Command-Line Format	<code>--mode</code>	
Permitted Values (Linux)	Type	string
	Default	<code>gtk</code>
	Valid Values	<code>gtk</code> (GTK (X Windows))
		<code>xwindow</code> (X Windows (native))
<code>text</code> (Text (command-line))		
	<code>unattended</code> (Unattended (no dialogs/prompts))	
Permitted Values (OS X)	Type	string
	Default	<code>osx</code>
	Valid Values	<code>osx</code> (Mac OS X (native))
		<code>text</code> (Text (command-line))
<code>unattended</code> (Unattended (no dialogs/prompts))		
Permitted Values (Unix)	Type	string
	Default	<code>xwindow</code>
	Valid Values	<code>xwindow</code> (X Windows (native))
		<code>text</code> (Text (command-line))
<code>unattended</code> (Unattended (no dialogs/prompts))		
Permitted Values (Windows)	Type	string
	Default	<code>win32</code>
	Valid Values	<code>win32</code> (Windows (native))
		<code>unattended</code> (Unattended (no dialogs/prompts))

The installation mode to use for this installation.

- `--debugtrace`

Command-Line Format	<code>--debugtrace</code>	
Permitted Values	Type	string

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

- `--debuglevel`

Command-Line Format	<code>--debuglevel</code>	
Permitted Values	Type	numeric
	Default	2 78

	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	<code>--installer-language</code>	
Permitted Values	Type	string
	Default	<code>en</code>
	Valid Values	<code>en</code> (English) <code>ja</code> (Japanese)

The installer language.

- `--installdir`

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

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	<code>--system-size</code>	
Permitted Values	Type	string
	Default	<code>medium</code>
	Valid Values	<code>small</code> (5 to 10 MySQL Servers monitored from a laptop or low-end server with no more than 4GB of RAM.)
		<code>medium</code> (Up to 100 MySQL Servers monitored from a medium-sized, but shared, server with 4 to 8GB of RAM.)
<code>large</code> (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	<code>--tomcatport</code>	
Permitted Values	Type	numeric
	Default	<code>18080</code> ⁷⁹

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	<code>--tomcatsslport</code>	
Permitted Values	Type	numeric
	Default	18443

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	<code>--mysql-identity-source</code>	
Permitted Values	Type	string
	Default	default
	Valid Values	default (Default) host_plus_datadir (host_plus_datadir)

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	<code>--mysql-ssl</code>	
Permitted Values	Type	boolean
	Default	0
	Valid Values	0 (Do not use SSL when connecting to the database) 1 (Use SSL when connecting to the database)

Use SSL when connecting to the database.

- `--adminuser`

Command-Line Format	<code>--adminuser</code>	
Permitted Values	Type	string
	Default	service_manager

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`

Command-Line Format	<code>--unattendedmodeui</code>	
Permitted Values	Type	string
	Default	<code>none</code>
	Valid Values	<code>none</code> (No dialogs)
		<code>minimal</code> (Critical dialogs)
	<code>minimalWithDialogs</code> (Minimal UI with dialogs)	

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

- `--adminpassword`

Command-Line Format	<code>--adminpassword</code>	
Permitted Values	Type	string

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

- `--mysql_installation_type`

Command-Line Format	<code>--mysql-installation-type</code>	
Permitted Values	Type	string
	Default	<code>bundled</code>
	Valid Values	<code>bundled</code> (Use the bundled MySQL server)
		<code>existing</code> (Use an existing (user supplied) MySQL server)

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

- `--dbport`

Command-Line Format	<code>--dbport</code>	
Permitted Values	Type	numeric
	Default	<code>13306</code>

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.

- `--dbhost`

Command-Line Format	<code>--dbhost</code>	
Permitted Values	Type	string
	Default	<code>127.0.0.1</code>

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	<code>--dbname</code>	
Permitted Values	Type	string
	Default	<code>mem</code>

The name of the MySQL Enterprise Service Manager repository.

- `--forceRestart`

Command-Line Format	<code>--forceRestart</code>	
Permitted Values	Type	boolean
	Default	<code>0</code>
	Valid Values	<code>0</code> (Do not restart services)
		<code>1</code> (Restart services)

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

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

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

- `--agentpassword`

Command-Line Format	<code>--agentpassword</code>	
Permitted Values	Type	string

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

- `--createBackup`

Command-Line Format	<code>--createBackup</code>	
Permitted Values	Type	boolean
	Default	1

Whether to backup the data.



Note

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

- `--restartImmediately`

Command-Line Format	<code>--restartImmediately</code>
----------------------------	-----------------------------------

Permitted Values	Type	boolean
	Default	1

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	<code>--agentuser</code>	
Permitted Values	Type	string
	Default	<code>agent</code>

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

- `--checkmysqlhost`

Command-Line Format	<code>--checkmysqlhost</code>	
Permitted Values	Type	string
	Default	<code>yes</code>
	Valid Values	<code>yes</code> (Check host)
		<code>no</code> (Do not check host)

Validate the MySQL hostname or IP address

- `--debuglevel`

Command-Line Format	<code>--debuglevel</code>	
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	<code>--debugtrace</code>	
Permitted Values	Type	string

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

- `--installdir`

Command-Line Format	<code>--installdir</code>	
Permitted Values (OS X)	Type	string
	Default	<code>/Applications/mysql/enterprise/agent/</code>
Permitted Values (Unix)	Type	string

	Default	/opt/mysql/enterprise/agent/
Permitted Values (Windows)	Type	string
	Default	C:\Program Files\MySQL\Enterprise\Agent

Specify the directory into which to install the software.

- `--installer-language`

Command-Line Format	<code>--installer-language</code>	
Permitted Values	Type	string
	Default	en
	Valid Values	en (English) ja (Japanese)

Set the language to use for the installation process.

- `--managerhost`

Command-Line Format	<code>--managerhost</code>	
Permitted Values	Type	string

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

- `--managerport`

Command-Line Format	<code>--managerport</code>	
Permitted Values	Type	numeric
	Default	18443

Tomcat SSL Port

- `--mode`

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

		<code>unattended</code> (Unattended (no dialogs/prompts))
Permitted Values (Windows)	Type	string
	Default	<code>win32</code>
	Valid Values	<code>win32</code> (Windows (native))
<code>unattended</code> (Unattended (no dialogs/prompts))		

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`

Command-Line Format	<code>--mysqlconnmethod</code>	
Permitted Values	Type	string
	Default	<code>tcpip</code>
	Valid Values	<code>tcpip</code> (Use TCP/IP)
<code>socket</code> (Use Unix Socket/Named Pipe)		

Specify the connection method to use to connect to MySQL. If you specify `tcpip`, the value of the `mysqlport` option 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`

Command-Line Format	<code>--mysqlhost</code>	
Permitted Values	Type	string
	Default	<code>127.0.0.1</code>

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

- `--mysqlpassword`

Command-Line Format	<code>--mysqlpassword</code>	
Permitted Values	Type	string

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

- `--mysqlport`

Command-Line Format	<code>--mysqlport</code>	
Permitted Values	Type	numeric
	Default	<code>3306</code>

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

- `--mysqlsocket`

Command-Line Format	<code>--mysqlsocket</code>	
----------------------------	----------------------------	--

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

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

- `--mysqluser`

Command-Line Format	<code>--mysqluser</code>	
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	<code>--agent-autocreate</code>	
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	<code>--generaluser</code>	
Permitted Values	Type	string

The username for the general user.

- `--generalpassword`

Command-Line Format	<code>--generalpassword</code>	
Permitted Values	Type	string

Password for the `--generaluser`.

- `--limiteduser`

Command-Line Format	<code>--limiteduser</code>	
Permitted Values	Type	string

The username for the limited user.

- `--limitedpassword`

Command-Line Format	<code>--limitedpassword</code>	
Permitted Values	Type	string

Password for the `--limitedpassword`.

- `--optionfile <optionfile>`

Command-Line Format	<code>--optionfile</code>	
----------------------------	---------------------------	--

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

- `--unattendedmodeui`

Command-Line Format	<code>--unattendedmodeui</code>	
Permitted Values	Type	string
	Default	<code>none</code>
	Valid Values	<code>none</code> (No dialogs)
		<code>minimal</code> (Critical dialogs)
	<code>minimalWithDialogs</code> (Minimal UI with dialogs)	

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

- `--use-external-glib`

Removed	3.0.14	
Permitted Values	Type	string

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`

Command-Line Format	<code>--version</code>
----------------------------	------------------------

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

- `--agent_installtype`

Command-Line Format	<code>--agent-installtype</code>	
Permitted Values	Type	string
	Default	<code>database</code>

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`

Removed	3.0.14	
Command-Line Format	<code>--ignore-old-proxy-aggr</code>	
Permitted Values	Type	boolean
	Default	<code>0</code>

Ignores Proxy and Aggregator while running an upgrade.

**Note**

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

- `--mysqlconnectiongroup`

Command-Line Format	<code>--mysqlconnectiongroup</code>	
Permitted Values	Type	string

Optionally sets the MySQL instance group for the connection.

As of 3.0.5, multiple groups can be assigned in a single installation by passing in a comma-separated list of group names.

- `--agentservicename`

Command-Line Format	<code>--agentservicename</code>	
Permitted Values (Linux)	Type	string
	Default	<code>mysql-monitor-agent</code>
Permitted Values (OS X)	Type	string
	Default	<code>mysql.monitor.agent</code>
Permitted Values (Unix)	Type	string
	Default	<code>mysql-monitor-agent</code>
Permitted Values (Windows)	Type	string
	Default	<code>MySQL Enterprise Monitor Agent</code>

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	<code>--help</code>
----------------------------	---------------------

Display the list of valid options to the installer.

Chapter 9 Performance Tuning MySQL Enterprise Monitor

Table of Contents

9.1 Tuning Memory	91
9.2 Tuning CPU	92
9.3 Tuning Apache Tomcat Threads	93
9.4 Tuning Agent Memory Requirements	94

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 9.1 Apache Tomcat configuration file location (default)

Operating System	Path
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\bin\setenv.bat</code>
Linux / Solaris	<code>/opt/mysql/enterprise/monitor/apache-tomcat/bin/setenv.sh</code>
Mac OS X	<code>/Applications/mysql/enterprise/monitor/apache-tomcat/bin/setenv.sh</code>

The following `setenv` variables are defined by the installation type:

Table 9.2 Installation Parameters

Parameter	Small	Medium	Large
Tomcat Heap Size	256MB	768MB	2048MB
Tomcat MaxPermSize	200MB	512MB	1024MB

- `--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.

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.

- `MaxPermSize`: defines the maximum size of the pool containing the data used by Tomcat's JVM.

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=-Jvms 768 --Jvmmx 768 ++JvmOptions="-XX:+UseParallelOldGC"
++JvmOptions="-XX:+HeapDumpOnOutOfMemoryError" ++JvmOptions="-XX:HeapDumpPath=@@BITROCK_TOMCAT_ROOTDIR@@"
++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 9.3 MEM repository configuration file location (default)

Operating System	Path
Microsoft Windows	C:\Program Files\MySQL\Enterprise\Monitor\mysql\my.ini
Linux / Solaris	home/mysql/enterprise/monitor/mysql/my.cnf
Mac OS X	/Applications/mysql/enterprise/monitor/mysql/my.cnf

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)

Operating System	Path
Microsoft Windows	C:\Program Files\MySQL\Enterprise\Monitor\bin\config.bat
Linux / Solaris	/opt/mysql/enterprise/monitor/bin/config.sh
Mac OS X	/Applications/mysql/enterprise/monitor/bin/config.sh

This Service Manager repository tool has the following options:

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

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)

Operating System	Path
Microsoft Windows	C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\conf\server.xml
Linux / Solaris	/opt/mysql/enterprise/monitor/apache-tomcat/conf/server.xml
Mac OS X	/Applications/mysql/enterprise/monitor/apache-tomcat/conf/server.xml

The following section should be modified:

```
[Redacted configuration snippet]
```

```
<Connector port="18443"  
  protocol="org.apache.coyote.http11.Http11Protocol" SSLEnabled="true"  
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"  
  ...
```

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: [mysqlnetworkctrl 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:

```
shell> ./uninstall
```

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.

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

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

Option	Description
<code>--help</code>	Displays the list of options.
<code>--version</code>	Displays the product name and version.
<code>--debuglevel <debuglevel></code>	Sets the verbosity of the uninstallation log. 0 is the lowest, 4 the highest, and 2 is the default.
<code>--mode <mode></code>	<p>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>.</p> <p>The following is a list of the GUI-based uninstallation options available:</p> <ul style="list-style-type: none"> • Windows: <code>Win32</code> • OS X: <code>osx</code> • Solaris: <code>xwindow</code> • Linux: <code>gtk</code> (Default) and <code>xwindow</code>. <p><code>--mode</code> can also initiate text mode and unattended uninstallations.</p> <ul style="list-style-type: none"> • <code>--mode text</code>: starts a text-only, console-based uninstallation process. Text-based uninstallation is not available on Windows platforms. • <code>--mode unattended</code>: starts an unattended uninstallation.
<code>--debugtrace <debugtrace></code>	Sets the path and filename of the uninstallation log file.
<code>--installer-language</code>	<p>Sets the language of the uninstallation. Possible values are:</p> <ul style="list-style-type: none"> • <code>en</code>: English. Default value. • <code>ja</code>: Japanese.

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:

```
shell>./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.

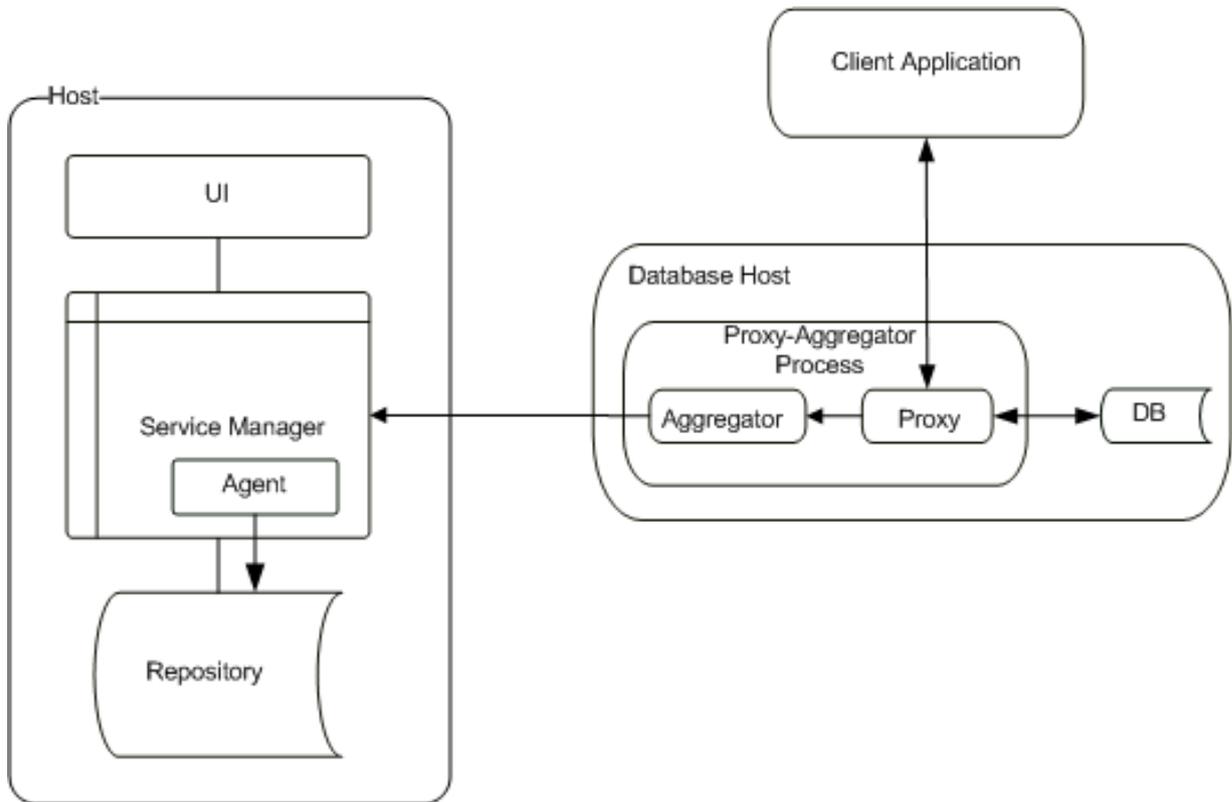
11.1 Proxy Aggregator Architecture

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.

Figure 11.1 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

that user, usually the `mysql` user. The same is true for the MySQL Enterprise Monitor Aggregator installation.

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';
```

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:



Note

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

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.

4. The Proxy installation configuration is completed. The MySQL Enterprise Monitor Aggregator installation configuration is described in [Installing the Aggregator](#).

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 11.1 MySQL Enterprise Monitor Proxy and Aggregator Installer Options

Option	Description
<code>--help</code>	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.
<code>--version</code>	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.
<code>--debuglevel</code>	Sets the verbosity of the installation log. 0 is the lowest verbosity, 4 is the highest. Default value is 2.
<code>--debugtrace</code>	Sets the path and filename of the installation log file.
<code>--optionfile</code>	Sets the path and filename of the installation options file. For more information, see Unattended Installation with Options File .
<code>--installer-language</code>	Sets the language of the installation. Possible values are: <ul style="list-style-type: none"> • <code>en</code>: English. Default value. • <code>ja</code>: Japanese.
<code>--mode</code>	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: <ul style="list-style-type: none"> • Windows: <code>Win32</code> • OS X: <code>osx</code> • Solaris: <code>xwindow</code> • Linux: <code>gtk</code> (Default) and <code>xwindow</code>.

Option	Description
	<p><code>--mode</code> can also initiate text mode and unattended installations.</p> <ul style="list-style-type: none"> • <code>--mode text</code>: starts a text-only, console-based installation process. Text-based installation are not available on Windows platforms. • <code>--mode unattended</code>: starts an unattended installation. For more information on unattended installations, see Unattended Installation from the Command Line and Unattended Installation with Options File.
<code>--unattendedmodeui</code>	<p>Sets the graphical elements to use, if any, in the unattended installation. The following options are available:</p> <ul style="list-style-type: none"> • <code>none</code>: no pop-ups, or progress bars are displayed. Errors are displayed, if they occur. • <code>minimal</code>: No user interaction is required and a progress bar is displayed showing the installation progress. Errors are displayed, if they occur. • <code>minimalWithDialogs</code>:
<code>--installdir</code>	Sets the installation directory for the product.
<code>--use-external-glib</code>	Sets the glib to use, the one delivered in the installer (0, default), or the system glib (1).
<code>--monitorcomponent</code>	<p>Specifies which component to install. The following options are available:</p> <ul style="list-style-type: none"> • <code>proxy</code>: installs both Proxy and Aggregator. This is the default. • <code>aggregator</code>: installs the Aggregator only.
Proxy-Specific Options	
<code>--proxyservicename</code>	Sets the unique service name for the Proxy service.
<code>--mysqlconnmethod</code>	<p>Sets the connection method used by the proxy to connect to the monitored MySQL instance. The following options are available:</p> <ul style="list-style-type: none"> • <code>tcpip</code>: default value. • <code>socket</code>: unavailable on Microsoft Windows platforms.
<code>--proxyport</code>	Sets the port the Proxy listens on for incoming connections. Default value is <code>6446</code> .
<code>--mysqlhost</code>	Sets the hostname or IP address of the monitored MySQL instance. Default value is <code>localhost</code> .
<code>--mysqlport</code>	Sets the port of the monitored MySQL instance. Default value is <code>3306</code> .
<code>--mysqlsocket</code>	Sets the socket used by the monitored MySQL instance.
Aggregator-specific Options	
<code>--aggregatorservicename</code>	Sets the unique name for the Aggregator service. Default value is <code>mysql-monitor-aggregator</code> .

Option	Description
<code>--aggregatorport</code>	Sets the port the Aggregator listens on. Default value is <code>14000</code> .
<code>--aggregatorsslport</code>	Sets the SSL port the Aggregator listens on for secure connections. Default value is <code>14443</code> .
<code>--aggregatorsslcertfile</code>	Sets the location of the SSL certificate.
<code>--aggregatorsslcafile</code>	Sets the location of the SSL CA file.
<code>--managerhost</code>	Sets the hostname or IP address of the MySQL Enterprise Service Manager installation. Default value is <code>localhost</code> .
<code>--managerport</code>	Sets the SSL port number of the MySQL Enterprise Service Manager's Tomcat installation. Default value is <code>18443</code> .
<code>--agentuser</code>	Sets the agent username which the Aggregator uses to communicate with the MySQL Enterprise Service Manager. Default value is <code>agent</code> .
<code>--agentpassword</code>	Sets the password of the agent used by the Aggregator.

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:

```
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

Option Name	Description
<code>-h, --help</code>	Lists the basic help options.
<code>--help-all</code>	Lists all available help options.
<code>--help-aggr</code>	Lists the Aggregator-specific help options.
<code>--help-proxy</code>	Lists the Proxy-specific help options. This option is only available on the <code>mysql-monitor-proxy</code> file.

Table 11.3 Application Options

Option Name	Description
<code>-V, --version</code>	Shows the version of the Proxy or Aggregator, depending on which file it is run with.
<code>--defaults-file=<file></code>	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 <code>--</code> prefix for each option) and call the file as needed.
<code>--verbose-shutdown</code>	Configures the application to always log the exit code on shutdown.
<code>--daemon</code>	Configures the application to run in daemon mode.
<code>--user=<user></code>	Defines the specific user to run the Aggregator.
<code>--basedir=<absolute path></code>	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.
<code>--pid-file=<file></code>	Defines the name of the PID file to use in the event the application is started in daemon mode.
<code>--plugin-dir=<path></code>	Defines the path to the plugins.
<code>--plugins-name=<name></code>	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.
<code>--log-level=<string></code>	Defines the logging level. Possible values are <code>critical</code> (default value), <code>error</code> , <code>warning</code> , <code>info</code> , <code>message</code> , and <code>debug</code> .
<code>--log-file=<filename></code>	Defines the name of the logfile.
<code>--log-use-syslog</code>	Configures the application to send all messages to the syslog. UNIX/Linux only.
<code>--log-backtrace-on-crash</code>	Configures the application to invoke the debugger in the event of a crash.
<code>--keepalive</code>	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.
<code>--max-open-files</code>	Configures the maximum number of open files.
<code>--event-threads</code>	Configures the number of event-handling threads. Default value is 1.
<code>--lua-path=<path></code>	Sets the <code>LUA_PATH</code> .
<code>--lua-cpath=<path></code>	Sets the <code>LUA_CPATH</code>

Table 11.4 aggr-module Options

Option Name	Description
<code>--aggr-address=<host:port></code>	Defines the address and listening port of the Aggregator. The default port value is 14000.
<code>--aggr-lua-script=<filename></code>	Defines the path to the LUA script.
<code>--aggr-mem-url=<url></code>	Defines the URL to the MySQL Enterprise Service Manager.
<code>--aggr-mem-user=<string></code>	Defines the Agent username to use for communication with the MySQL Enterprise Service Manager.
<code>--aggr-mem-password=<string></code>	Defines the Agent password to use for communication with the MySQL Enterprise Service Manager.
<code>--aggr-ssl-address=<host:port></code>	Defines the address and listening port of the Aggregator for SSL connections to the Aggregator.
<code>--aggr-ssl-cert-file=<filename></code>	Defines the PEM server certificate for the Aggregator.
<code>--aggr-ssl-cs-file=<filename></code>	Defines the CA certificate for the Aggregator.
<code>--aggr-ssl-ciphers=<string></code>	Defines the supported ciphers.
<code>--aggr-test-mode</code>	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.
<code>--aggr-flush-interval=<seconds></code>	Defines the interval, in seconds, at which the query data is flushed to the MySQL Enterprise Service Manager. The default value is 60 seconds.
<code>--aggr-max-request-body-size=<bytes></code>	Defines the maximum size of an HTTP request body. The default size is 1MB.

Table 11.5 proxy-module Options

Option Name	Description
<code>-P, --proxy-address=<host:port></code>	The address and listening port of the Proxy. Default port is 4040.
<code>-r, --proxy-read-only-backend-addresses</code>	The address and listening port of the remote, slave server. This is not set by default.
<code>-b, --proxy-backend-addresses=<host:port></code>	<p>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.</p> <p>For example, if you specify two servers A and B, the first client connection goes to server A; the second client connection to server B and the third client connection to server A. When using this option on the command line, you can specify the option and the server multiple times to specify multiple backends.</p> <p>When using this option in a configuration file, separate multiple servers with commas.</p>
<code>--proxy-skip-profiling</code>	Disable query profiling (statistics time tracking). The default is for tracking to be enabled.
<code>-s file_name, --proxy-lua-script=<file></code>	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

Option Name	Description
	reloaded for each connection; if the content of the Lua script changes while the Proxy is running, the updated content is automatically used when a new connection is made.
<code>--no-proxy</code>	Disables the Proxy module. By default, the Proxy is enabled.
<code>--proxy-pool-no-change-user</code>	Disable use of the MySQL protocol <code>CHANGE_USER</code> command when reusing a connection from the pool of connections specified by the <code>proxy-backend-addresses</code> list.
<code>--proxy-connect-timeout</code>	Defines the Proxy's connection timeout in seconds. Default value is 2.
<code>--proxy-read-timeout</code>	Defines the read timeout in seconds. Default is 8 hours.
<code>--proxy-write-timeout</code>	Defines the write timeout in seconds. Default is 8 hours.

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:

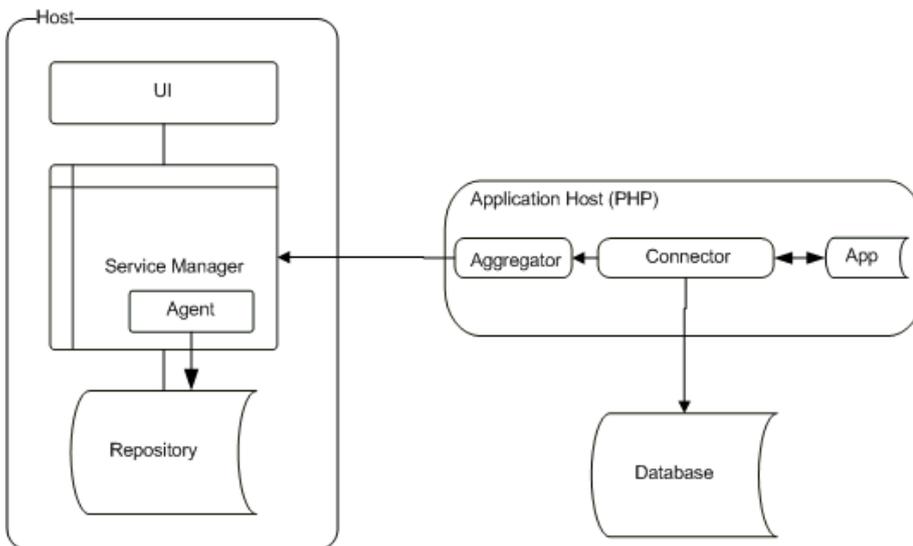
- [Section 12.1, “Using the MySQL Enterprise Plugin for Connector/PHP”](#): describes the configuration of a PHP-based application which uses the MySQL Enterprise Plugin for Connector/PHP and MySQL Enterprise Monitor Aggregator to feed query information to the Query Analyzer
- [Section 12.2, “Using the 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

Enterprise Service Manager. The other Connectors can be configured to do this without the MySQL Enterprise Monitor Aggregator.

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:

```
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>.

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 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 `.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, or `php_mysqlenterprise_ts.dll` on Microsoft Windows. If not, use `mysqlenterprise.so` on Linux, Unix, and OS X, or `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:

```
extension=/path/to/mysqlenterprise.so
```

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:

```
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:

```
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:

```
$ 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

Property	Description
<code>aggregator_connect_timeout_sec</code>	<p>Timeout, in seconds, for communications with the MySQL Enterprise Monitor Aggregator.</p> <ul style="list-style-type: none"> Property type: <code>integer</code> Default value: <code>1</code> <p>This property can be combined with the <code>aggregator_connect_timeout_usec</code> property.</p>
<code>aggregator_connect_timeout_usec</code>	<p>Timeout, in microseconds, for communications with the MySQL Enterprise Monitor Aggregator.</p> <ul style="list-style-type: none"> Property type: <code>integer</code> Default value: <code>0</code> <p>This property can be combined with the <code>aggregator_connect_timeout_sec</code> property.</p>
<code>aggregator_user</code>	<p>The Aggregator's username. See Chapter 11, Proxy and Aggregator Installation for more information.</p> <ul style="list-style-type: none"> Property type: <code>string</code>
<code>aggregator_password</code>	<p>The Aggregator's password.</p> <ul style="list-style-type: none"> Property type: <code>string</code>
<code>aggregator_url</code>	<p>The IP address, or hostname, and port of the Aggregator installation.</p> <ul style="list-style-type: none"> Property type: <code>string</code> Default value: <code>tcp://127.0.0.1:14000</code>
<code>debug_callback</code>	<p>This property should be used only when debugging your MySQL Enterprise Monitor installation with MySQL Support personnel.</p> <p>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.</p>
<code>disable_backtrace</code>	<p>Defines whether a backtrace is performed. Backtrace is useful for debugging but has a performance impact.</p> <ul style="list-style-type: none"> Property type: <code>boolean</code> Default value: <code>1</code>, the backtrace is disabled. To enable the backtrace, set this value to <code>0</code>.
<code>log_file</code>	<p>Defines the location of a log file which logs all query information sent to the Aggregator.</p>

Property	Description
	This should only be used for debugging purposes because every request is logged, resulting in a very large log file. <ul style="list-style-type: none"> Property type: <code>string</code>
<code>quan_enabled</code>	Defines whether query analysis is enabled. <ul style="list-style-type: none"> Property type: <code>boolean</code> Default value: <code>1</code>, query analysis is enabled. To disable query analysis, set this value to <code>0</code>.

The following is an example of the Aggregator-specific section of the `php.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

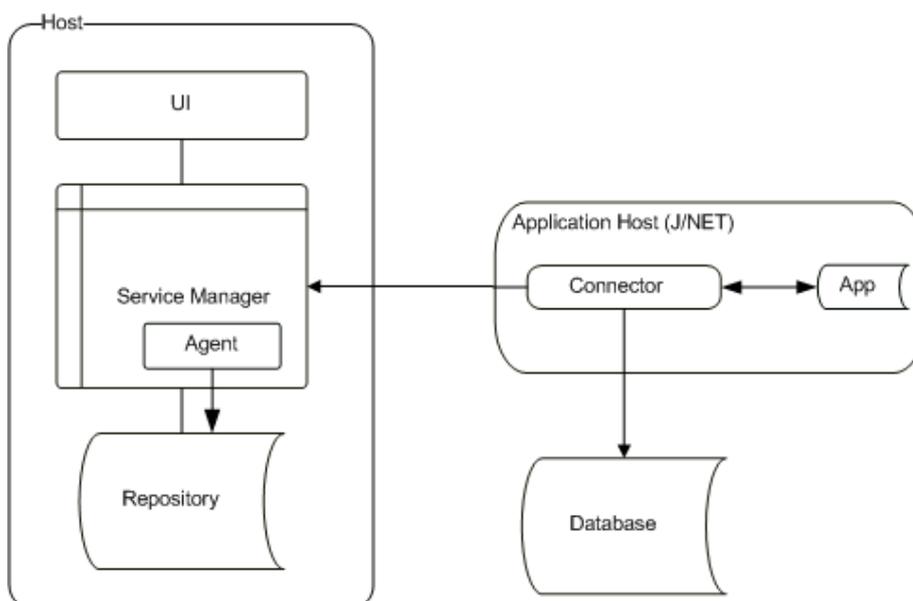
The MySQL Enterprise Plugin for Connector/J, enables query analysis for your applications without requiring any modification to the application code.



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>
  <listener-class>
    com.mysql.etools.jdbc.ContextListener
  </listener-class>
</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

Property Name	Description
<code>statementInterceptors</code>	Enables the plugin. Set this property as follows: <code>statementInterceptors = com.mysql.etools.jdbc.StatementPerformanceCount</code>

Property Name	Description
<code>disableSourceLocation</code>	<p>Defines whether to send stack traces with example queries to MySQL Enterprise Service Manager.</p> <ul style="list-style-type: none"> Property type: <code>boolean</code> Values: <code>true</code> or <code>false</code> (default).
<code>serviceManagerUrl</code>	<p>Defines the URL of the MySQL Enterprise Service Manager. Include the full URL and port number.</p> <ul style="list-style-type: none"> Property type: <code>string</code> Value: URL and port number of MySQL Enterprise Service Manager.
<code>serviceManagerUser</code>	<p>Defines the Agent username to use when connecting to MySQL Enterprise Service Manager.</p> <ul style="list-style-type: none"> Property type: <code>string</code>
<code>serviceManagerPassword</code>	<p>Defines the Agent password to use when connecting to MySQL Enterprise Service Manager.</p> <ul style="list-style-type: none"> Property type: <code>string</code>
<code>serviceManagerConnectTimeout</code>	<p>Defines the number of seconds to wait for a connection to MySQL Enterprise Service Manager.</p> <ul style="list-style-type: none"> Property type: <code>numeric</code> Default value: 0
<code>serviceManagerResponseTimeout</code>	<p>Defines the number of seconds to wait for a response from MySQL Enterprise Service Manager.</p> <ul style="list-style-type: none"> Property type: <code>numeric</code> Default value: 0
<code>mysqlServerUUID</code>	<p>If you are unable to retrieve the server's UUID, define it with this property.</p> <ul style="list-style-type: none"> Property type: <code>string</code> <p>To retrieve the UUID, the plugin requires <code>SELECT</code> privileges on <code>mysql.inventory</code>.</p>

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

Property Name	Description
<code>verifySslHostnames</code>	<p>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.</p>

Property Name	Description
	<ul style="list-style-type: none"> Property type: <code>boolean</code> Default value: <code>false</code>
<code>verifySslCerts</code>	<p>Defines whether the plugin verifies the certificate presented by the server was signed by a CA in the <code>trustCertificateKeystore</code>.</p> <ul style="list-style-type: none"> Property type: <code>boolean</code> Default: <code>false</code>, verification is disabled.
<code>trustCertificateKeystoreUrl</code>	<p>Defines the URL of the trusted root certificate KeyStore. If none is specified, the Java defaults are used.</p> <ul style="list-style-type: none"> Property type: <code>string</code>
<code>trustCertificateKeystorePassword</code>	<p>Defines the password for the KeyStore.</p> <ul style="list-style-type: none"> Property type: <code>string</code>
<code>trustCertificateKeystoreType=[type]</code>	<p>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.</p>
<code>clientCertificateKeystoreUrl</code>	<p>Defines the URL of the client KeyStore. If none specified, Java defaults are used.</p>
<code>clientCertificateKeystorePassword=[password]</code>	<p>Defines the password to use for the client certificate store.</p>
<code>clientCertificateKeystoreType</code>	<p>Defines the KeyStore type for client certificates. If type is set to NULL or empty, JKS is used by default.</p>

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:

```
statementInterceptors=com.mysql.etoools.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:

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

Connection conn = null;
...
try {
    conn =
        DriverManager.getConnection("jdbc:mysql://localhost/test?" +
```

```

        "?user=mysqltest" +
        "&password=mysqltest" +
        "&statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters" +
        "&serviceManagerUrl=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)

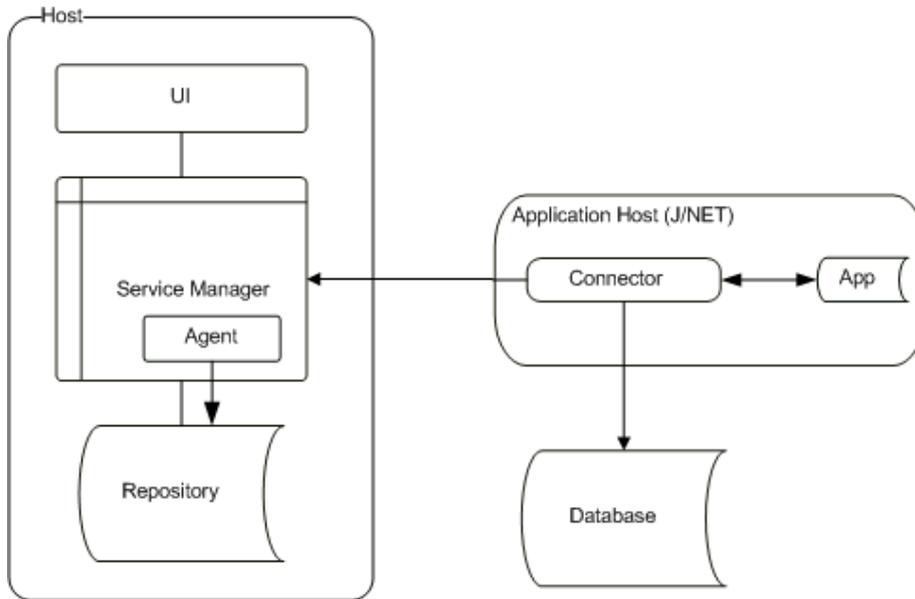
Additional configuration of the plugin is done in the Enterprise Service Manager's user interface, using the **Query Analyzer** tab. From here, the capture of query performance data for a given MySQL instance that the plugin is being with can be enabled or disabled. It is also possible to configure the 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.

Figure 12.3 Connector Plugin Architecture



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:

```
<system.diagnostics>
  <sources>
    <source name="mysql" switchName="SourceSwitch"
      switchType="System.Diagnostics.SourceSwitch">
      <listeners>
        <add name="EMTrace" type="MySql.EMTrace.EMTraceListener, MySql.MonitorPlugin"
          initializeData=""
          Host="SERVERHOST:SERVERPORT"
        />
      />
    />
  />
```

```

        PostInterval="POSTINTERVAL"
        UserId="AGENTUSERID"
        Password="AGENTPASSWORD" />
    </listeners>
</source>
</sources>
<switches>
    <!-- You can set the level at which tracing is to occur -->
    <add name="SourceSwitch" value="All" />
</switches>
</system.diagnostics>

<system.data>

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

</system.data>

```

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:

```

<trace autoflush="false" indentsize="4">
    <listeners>

```

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

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

21 GUI-Based Advisor Reference	221
21.1 Agent Health Advisor	221
21.2 MySQL Enterprise Backup Health Advisor	223
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
22 Customizing MySQL Enterprise Monitor	231
22.1 Customizing Groups	231
22.2 Creating Advisors and Rules	232
22.2.1 Creating Advisors	232
22.2.2 Overview of Graph Creation	233
22.2.3 Overview of Advisor Creation	234
22.2.4 Variables	235
22.2.5 Thresholds	235
22.2.6 Using Strings	236
22.2.7 Wiki Format	236
22.2.8 Creating a New Advisor: An Example	237
22.2.9 Creating a New Graph: An Example	239
22.3 Custom Data Collection	240
22.3.1 Custom.xml	240
22.3.2 Queries	241
22.3.3 Data Collection Attributes	242
22.4 Event Notification Blackout Periods	244
22.4.1 Scripting Blackouts	245

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.

**Note**

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

**Note**

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

**Note**

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

ORACLE MySQL Enterprise Monitor

1 2 0 0 0 manager

Dashboards Events Query Analyzer Reports & Graphs Configuration

Configure What's New

Welcome to MEM 3.0!

Some highlights and new features:

1. Easy to Use and Configure - install one Agent per host, then auto-discover and set up monitoring of your MySQL instances and host resources centrally from the Dashboard
2. Track database availability, problem Hosts & Instances, and analyze SQL response times
3. Get instant Query Analyzer results to tune your SQL performance when monitoring MySQL 5.6 or later (or download a plugin for earlier versions)
4. Explore MySQL run-time configuration and status (hosts tool)
5. Operating System and other metrics now include historical trend analysis, resource forecasting and predictive alerts
6. Replication analysis lets you identify and fix slave lag, outliers...
7. Rich graph package includes completely tuned graphs aggregated and individual graphs
8. Policy driven event handling & notifications

Read more...

New users - Guide to completing your configuration

Existing users - Guide to completing your upgrade

Important Product-Related Announcements

MySQL Enterprise Monitor 2.3.13 Is Now Available!

Feb 20, 2013 8:27:22 AM

We are pleased to announce that MySQL Enterprise Monitor 2.3.13 is now available for download on the My Oracle Support (MOS) web site. It will also be available via the Oracle Software Delivery Cloud with the March update in about 3 weeks. This is a maintenance release that contains several new features and fixes a number of bugs. There are 13 new rules and 6 new graphs focused mainly on new features in MySQL Server 5.6. You can find more information on the contents of this release in the change log:

<http://dev.mysql.com/doc/mysql-monitor/2.3/en/mem-news-2-3-13.html>

You will find binaries for the new release on My Oracle Support:

<https://support.oracle.com>

Choose the "Patches & Updates" tab, and then use the "Product or Family (Advanced Search)" feature.

Copyright © 2005, 2013, Oracle and/or its affiliates. All rights reserved. 3.0.0.2880 - scissors (192.168.1.215) - Aug 19, 2013 8:53:08 pm (Up Since: 3 hours ago) - About

The **What's New** page is divided into three main sections:

- **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

configure what's new SQL Enterprise Monitor

Automatically check for Product-Related Announcements?
Last Check: Aug 19, 2013 8:58:06 PM
Force check now? yes no

Automatically check for updates to open My Oracle Support Issues?
Last Check: never
Force check now? yes no

More information on [configuring and troubleshooting What's New.](#)

Save Cancel

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 **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

Name	Description
Group Overview	Drop-down list containing all groups defined in MySQL Enterprise Service Manager.

Name	Description
Graph Time Range	<p>Drop-down lists containing the time periods to apply to the graphs. The possible values are:</p> <ul style="list-style-type: none"> • Interval: select the duration for the overview data. If you select 1 hour, the data collected in the last hour is displayed. • From/To: select a date and time range for the overview data. <p>Graph time ranges apply to the timeseries graphs, only. They do not apply to the Availability, Problem, or Current Emergency information.</p>
Go	Applies the new configuration to the Overview dashboard.
Save as Default	<p>Sets the selected group and time range as the default.</p> <p>It is not possible to save a date range, using From/To, as the default for a group.</p>
Reset	Resets the group overview configuration to the previously saved values.



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

= 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 known 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

Name	Description
Agent Host	Lists the hostname of the monitoring agent.
Connection Details	Lists the IP address defined in the connection string.
Last Error Date	Date and time of the last occurrence of this error.
Error Details	Cause of the error.

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

Name	Description
Agent	Hostname of the server on which the agent is installed.
State	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 SHUTDOWN . If the agent did not shutdown properly, if its host shutdown unexpectedly, or due to a network fault, the state displayed is TIMEDOUT .
Last Seen	Time and date at which the agent last contacted the Service Manager.
Version	Agent version.
UUID	The unique identifier of the agent.
Agent Directory	Agent installation directory.

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

Name	Description
Host	The server on which the running MySQL instance was discovered.
Connecting	Whether a connection is being attempted with the instance.
Port/Socket	Port or socket on which the MySQL instance is listening.
Process ID	The process ID of the running instance.
Process User: Group	ID of the user and group.
Process Arguments	The arguments with which the instance was started.

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

Name	Description
Instance	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.
Notes	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.

Name	Description
Versions: MySQL	Displays the version of the monitored MySQL instance.
Versions: Agent	Displays the version of the monitoring agent.
Versions: Operating System	Displays the type and version of operating system on which the MySQL instance is installed.
Port	Displays the configured MySQL port.
Data Dir	Displays the configured data directory of the MySQL installation.

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

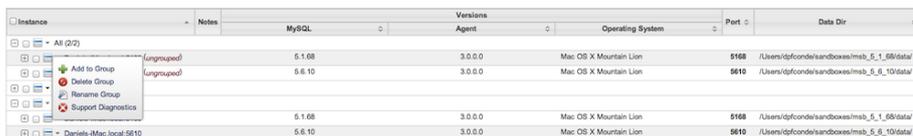
The Configuration schema may contain login credentials. However, these credentials are encrypted using keys which are not stored in the repository and are not included in the Support Diagnostics report.



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.



Important

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.

- **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

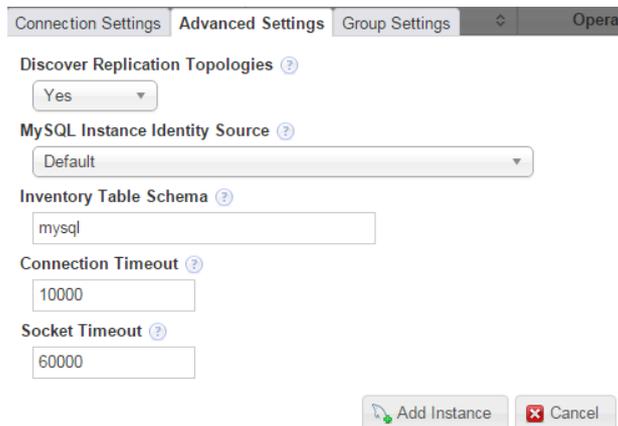
Table 14.6 Connection Settings Tab

Name	Description
Monitor From	Select an Agent from the list of Agents to monitor this MySQL database Instance.

Name	Description
	<p>It is recommend installing one Agent per Host and using that Agent to monitor all its MySQL Instances.</p> <p>Set up the Agent as a local connection by specifying TCP/IP and an Instance Address 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.</p> <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;">  </div> <div> <p>Note</p> <p>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.</p> </div> </div>
Connect Using	<p>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.</p>
Instance Address and Port	<p>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.</p>
Admin User and Password	<p>The root user, or user with the privileges defined in Creating the Admin User and the password.</p>
Auto-Create Less Privileged Users	<p>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".</p> <p>Choose No if you intend to use the Admin user for all data collection.</p> <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;">  </div> <div> <p>Note</p> <p>It is strongly recommend to use the General and Limited user.</p> </div> </div>
General User and Password	<p>Add a user name and password for the General User.</p>
Limited User and Password	<p>Add a user name and password for the Limited User.</p>

Advanced Settings

Figure 14.4 Add Instance Advanced Settings



Connection Settings | **Advanced Settings** | Group Settings | Opera

Discover Replication Topologies ?

MySQL Instance Identity Source ?

Inventory Table Schema ?

Connection Timeout ?

Socket Timeout ?

Table 14.7 Advanced Settings

Name	Description
Discover Replication Topologies	<p>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.</p> <p>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.</p> <p>If set to No, no replication discovery is attempted.</p>
MySQL Identity Source	<p>Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.</p> <ul style="list-style-type: none"> • Default: uses either the <code>server_uuid</code> variable if present, or generates a random new UUID • Host Plus Datadir generates a hash of the host identity and the path to the MySQL instances data directory to create a unique identity. <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"></div> <div> <p>Note</p> <p>Host Plus Datadir can be used only if the agent is running on the same host as the MySQL instance for this connection.</p> </div> </div>
Inventory Table Schema	<p>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.</p> <p>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.</p>
Connection Timeout	Connection timeout, in milliseconds, used by the JDBC driver.
Socket Timeout	Socket timeout, in milliseconds, used by the JDBC driver.

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.

Figure 14.5 MySQL Instance Filter



Table 14.8 MySQL Instance Filter

Name	Description
Server Name	Search on full or partial name of the server.
Server UUID	Search on the UUID of the server.
Server ID	Search on the <code>server_id</code> .
Query Analyzer	Search for servers on which the Query Analyzer is enabled, or not.
MySQL Version	Search for specific MySQL version numbers.
Agent Version	Search for specific MySQL Enterprise Monitor Agent version numbers.
Operating System	Search on the Operating Systems on which the server is installed.

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

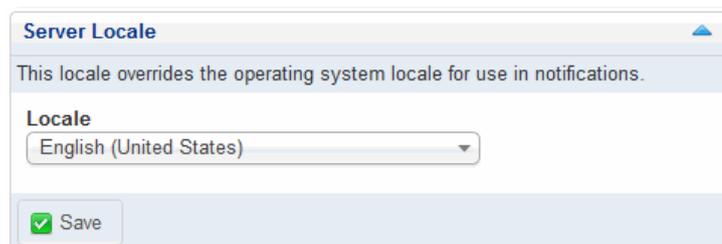
MySQL Enterprise Monitor is available as part of the MySQL Enterprise subscription, learn more at <http://www.mysql.com/products/>.

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.

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



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.

**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**

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

Customize MySQL server name

Customize how MySQL server names are displayed in the application. Typically, MySQL server names are displayed as concatenations of the server name and a connection endpoint value (where a connection endpoint can be a port number, socket etc.).

Show MySQL server names as:

- Reported by agent (Default)
- Hostname only
- Transformed by substitution expression

Substitution Expression

Example: The expression `mysql=oracle` transforms `host.mysql.com` to `host.oracle.com`

Display connection endpoint values:

- Always (Default)
- Never
- For non-default values

Save

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:

```
dx521\.\example\.\com=Staging
dx984[.\]example[.\]com=Production
database-server-(.*?)\.\example\.\com=$1
^database-server="--", [.\]example[.\]com$="--", dx521="Staging DB",
```

If you use incorrect syntax in the replacement expression, the original hostname is displayed with a suffix such as (!); this suffix varies depending on your locale setting.

- The **Display connection endpoint values** controls whether the detailed connection part of the hostname (such as the TCP/IP port number) is displayed. By default, it is always shown. You can set it to never be shown, or to only be shown if it is not the standard value. For TCP/IP connections, the default port is 3306. For socket connections, the default endpoint value is `/tmp/mysql.sock`.

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

User Login	Full Name	Role	Authentication	Password set date	Last Login	Failed logins	Last failed login
agent		agent	Built-in	Aug 19, 2013 6:49:53 PM		0	
manager		manager	Built-in	Aug 19, 2013 6:49:53 PM	Aug 22, 2013 11:17:50 AM	0	
john	John Doe	dba	Built-in	Aug 22, 2013 11:22:18 AM		0	
readonly		read-only	Built-in	Aug 22, 2013 11:22:54 AM		0	

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



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

Log Name	Threshold	Last Modified	Entries
All	N/A	Aug 22, 2013 11:17:52 AM	1,196
Advisors	Info	Aug 19, 2013 5:53:06 PM	393
Agent Monitoring	Info	N/A	0
Agent Tasks	Info	N/A	0
Apache	Warning	N/A	0
Catalina	Warning	N/A	0
Data Collection	Info	N/A	0
EH Cache	Warning	N/A	0
Freemarker	Warning	N/A	0
Graphs	Info	N/A	0
Groups	Info	N/A	0
Hibernate	Warning	Aug 19, 2013 9:23:07 PM	633
Infrastructure	Info	Aug 20, 2013 9:41:17 PM	5
Inventory	Info	N/A	0
JDBC	Info	N/A	0
Migration	Info	Aug 19, 2013 5:53:06 PM	2
Misc	Info	N/A	0
Network	Info	Aug 19, 2013 5:58:09 PM	4
Notification	Info	N/A	0
Preferences	Info	N/A	0

Copyright © 2005, 2013, Oracle and/or its affiliates. All rights reserved. 3.0.0.2880 - scissors (192.168.1.215) - Aug 22, 2013 11:17:59 am (Up Since: 2 days, 17 hours ago) - About

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/>.

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

Figure 15.9 MySQL Enterprise Monitor User Interface: User Preferences

The screenshot shows the 'User Preferences' page in the MySQL Enterprise Monitor interface. At the top, the Oracle logo and 'MySQL Enterprise Monitor' are visible. The user 'john' is logged in. The navigation menu includes 'Dashboards', 'Events', 'Query Analyzer', 'Reports & Graphs', and 'Configuration'. The 'User Preferences' section contains the following fields:

- User Name:** john
- Full Name:** John Doe
- Role:** dba
- Password (Leave empty to keep password unchanged):** [Empty]
- Confirm Password:** [Empty]
- Timezone:** (GMT-08:00) Pacific Standard Time - US/Pacific
- Locale:** English (United States)

There are 'Save' buttons with a checkmark icon at the top and bottom of the form. The footer contains the text: 'Copyright © 2005, 2013, Oracle and/or its affiliates. All rights reserved. 3.0.0.2880 - scissors (192.168.1.215) - Aug 22, 2013 11:26:27 am (Up Since: 2 days, 17 hours ago) - About'.

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.
- `com.mysql.ertools.agent.csv`: A built-in MySQL Enterprise Monitor Agent 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.log`: A general built-in MySQL Enterprise Monitor Agent log file.
- `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.

- `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.

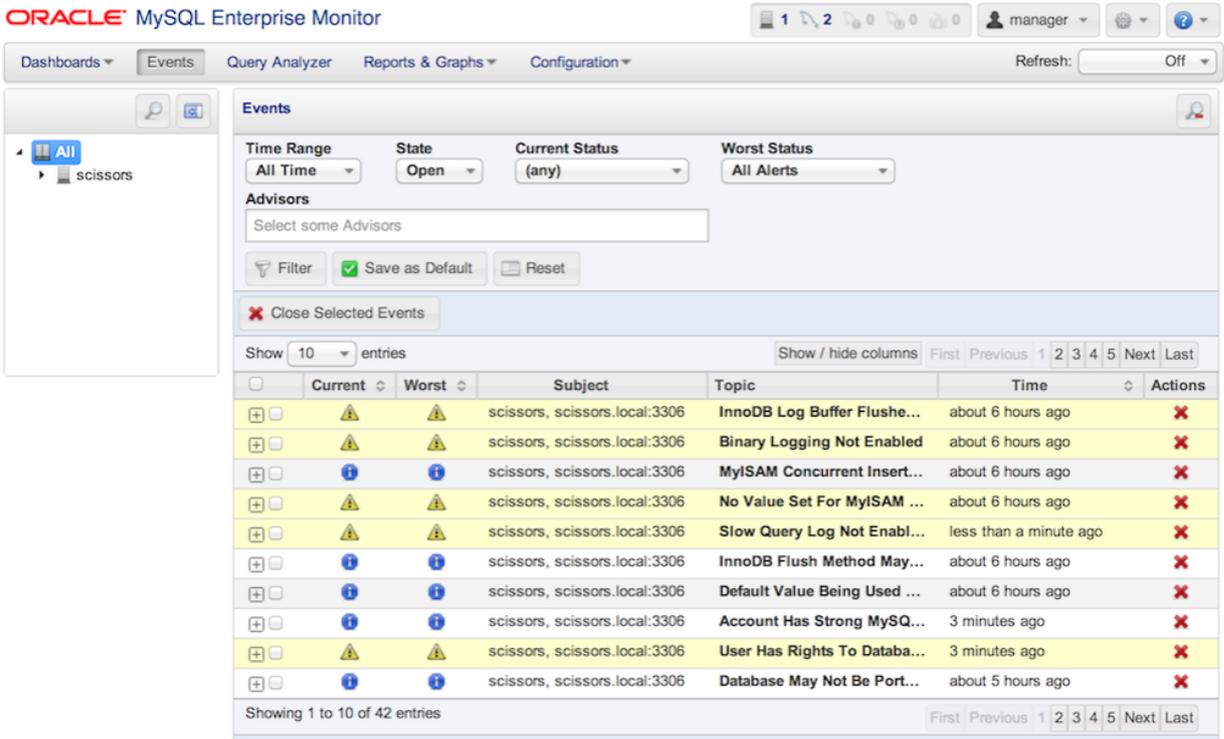
Icon	Description
	A red and orange flames icon indicates an emergency alert.
	An round red icon indicates a critical alert.

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 **Filter** 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.

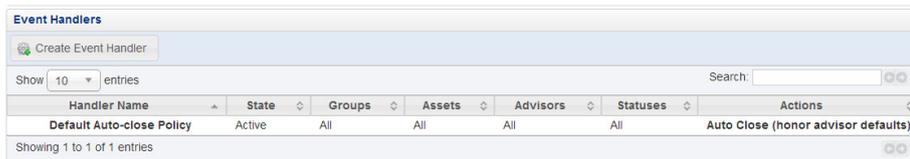
The **Event Handler** page is grouped in the following sections:

- **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

Name	Description
Create Event Handler	Opens the Create Event Handler dialog. For more information, see
Show * Entries	Select the maximum number of event handlers to display.

Name	Description
Handler Name	Lists the names of the event handlers.
State	Lists the state of the event handler. Possible states are: <ul style="list-style-type: none"> • Active: the event handler is running. • Suspended: the event handler is not running.
Groups	Lists the groups assigned to the event handler.
Assets	Lists the assets assigned to the event handler.
Advisors	Lists the Advisors assigned to the event handler.
Statuses	Lists the statuses assigned to the event handler.
Actions	Lists the SMTP or SNMP actions assigned to the event handler.
Search	Enables you to search for specific event handlers.

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 18.2 Email Notification Groups Controls

Name	Description
Create Notification Group button	Opens the Create Notification Group dialog. For more information, see
Group Name	Lists the names of the notification groups.
Recipients	Lists the recipients' email addresses.

Name	Description
Subject Line	The subject line of the notification emails.
SMS	Status of SMS encoding. The following values are possible: <ul style="list-style-type: none"> • true: SMS encoding is enabled. • false: SMS encoding is not enabled.
MEM Admin	Status of emails regarding MySQL Enterprise Monitor. The following values are possible: <ul style="list-style-type: none"> • true: critical MySQL Enterprise Monitor emails will be sent to this notification group. • false: no email related to MySQL Enterprise Monitor will be sent to this notification group.

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

The screenshot shows a dialog box titled "Create Group". It has the following elements:

- Group Name**: A text input field.
- Recipients (comma separated or one email address per line)**: A larger text area for entering email addresses.
- Subject Line (use {0} for alarm level, {1} for rule name, and {2} for server name)**: A text input field containing the example "MySQL Enterprise Monitor {0} Alert: {1} ({2})".
- SMS (use SMS encoding for this notification group)**: A checkbox that is currently unchecked.
- MEM Admin (send MySQL Enterprise Monitor critical system messages to this notification group)**: A checkbox that is currently unchecked.
- Buttons**: "Save Notification Group" (with a checkmark icon) and "Cancel" (with an X icon).

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

Name	Description
Enable Email Notifications	Select to activate the email settings controls.
From Address	The email address added to the From field of all emails sent from MySQL Enterprise Monitor.
SMTP server	The outgoing email server address
SMTP Server Login	The username for the SMTP server
Update Password on Save	Select to activate the password fields.
Disable JavaMail TLS/SSL	Select if the SMTP server does not require an encrypted connection.
On Save, Send Test Email Message To	Enter an email address if you want to send a test email when the changes are saved.
Save Email Settings	Saves the Email Settings and sends a test email if an address is defined in the On Save, Send Test Email Message To field.

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

Name	Description
Enable SNMP Notifications	Activates the SNMP configuration fields.
Use SNMP v1/v2	Choose the version of SNMP you intend to use.
Target and Port Number	IP address and Port number of the system which will receive the SNMP Traps.
Community String	SNMP community string. Default value is <code>public</code> .
Use the remote MySQL agent host IP address as the SNMP trap agent address for Advisor traps (optional)	<p>Defines the source IP address included in the trap.</p> <ul style="list-style-type: none"> • Disabled: the trap uses the IP address of the service manager. • Enabled: the trap uses the IP address of the agent monitoring the host for which the advisor was triggered.
SNMP trap agent address for internally generated traps (optional)	Defines the source IP address included in traps generated by MySQL Enterprise Service Manager
On Save send test trap	Send a test trap message when Save is clicked. Select one, or more, of the trap types from the list. One trap is sent for each option selected.

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:

Figure 18.6 Create Event Handler Dialog

Create Event Handler [REDACTED]

Event Handler Name ?

Filters

Assets ?

Groups ?

Advisors ?

Event Statuses ?

Event Handling

SMTP Notification Groups ?

SMTP Notification Policy ?

SMTP Rate Limit ?

Send SNMP Traps ?

SNMP Notification Policy ?

Auto-Close Events ?

Table 18.5 Create Event Handler Controls

Name	Description
Event Handler Name	Specify a name which uniquely identifies the new event handler.
Assets	<p>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.</p> <p>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.</p> <div style="display: flex; align-items: flex-start; margin-top: 10px;"> <div style="flex: 1; text-align: center; margin-right: 10px;">  </div> <div style="flex: 2;"> <p>Note</p> <p>If you select the top-level of the asset, all assets are selected. This includes network interfaces, file systems, MySQL instances, and so on. You must expand the asset's entry to select individual assets.</p> </div> </div>
Groups	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.
Advisors	Select the Advisors. If this field is left blank, all advisors are included in the event handler's condition.
Event Statuses	Select the statuses for which you want to receive notifications.

Name	Description
SMTP Notification Groups	Select the groups you want to notify.
SMTP/SNMP Notification Policy	<p>Select one of the following:</p> <ul style="list-style-type: none"> • Notify on event escalation: 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. • Notify on any status change: sends a notification if the status changes to any other status. • Always notify: 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. <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;">  </div> <div> <p>Note</p> <p>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.</p> </div> </div>
SMTP Rate Limit	<p>Defines the maximum number of SMTP notifications which can be sent per minute. The default is 40.</p> <p>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.</p> <p>These events are not auto-closed and are updated only for the first failure.</p>
Send SNMP Traps	Defines whether SNMP Traps are used for notifications.
Auto-Close Events	<p>Defines whether the events are closed after the trigger status changes. The following values are possible:</p> <ul style="list-style-type: none"> • Yes: the default auto-close policy is honored and the events are closed when the conditions defined are no longer met. • No: the default auto-close policy is ignored and the events remain open in the Events page even after the conditions are no longer met.



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:

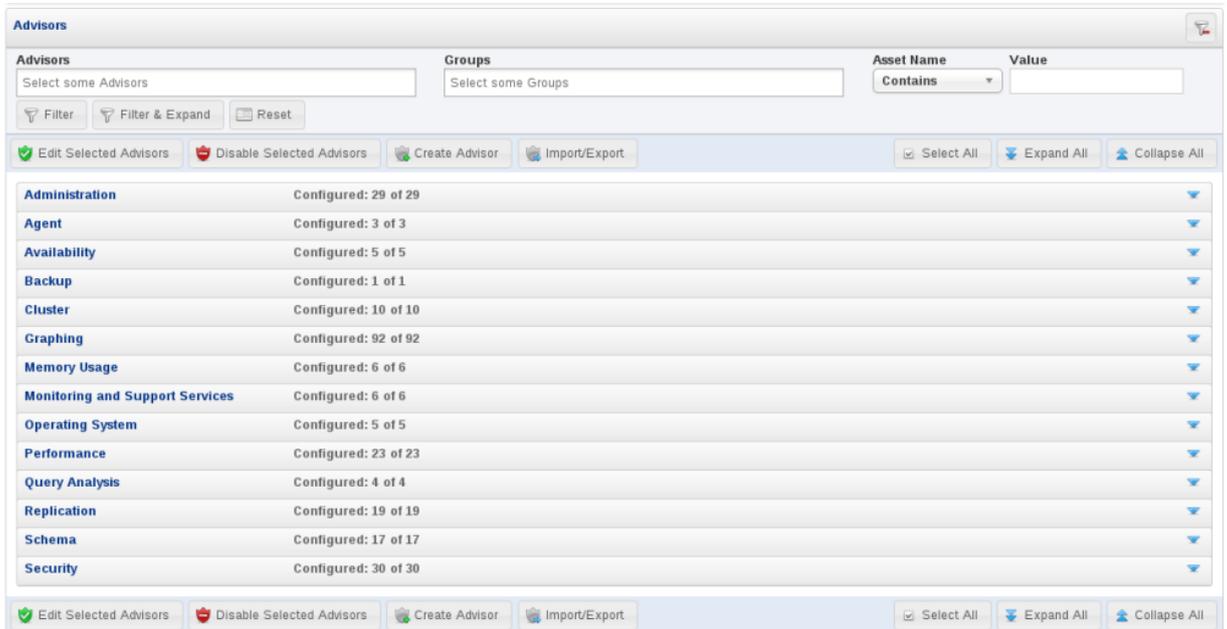
- [Section 19.1, “Advisors Page”](#)
- [Section 19.2, “Advisor Types”](#)
- [Section 19.3, “Advisor Thresholds”](#)
- [Section 19.4, “Advisor Schedules”](#)

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

Name	Description
Edit Selected Advisors	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

Name	Description
	for multiple Advisors simultaneously. You can also edit an advisor using the drop-down menu adjacent to each advisor's name.
Disable Selected Advisors	Disables all selected Advisors.
Create Advisor	Opens the Create Advisor page.
Import/Export	Opens the Custom Rule/Graph Export page. Note: This functionality is for custom rules and graphs only.
Select All	Selects all Advisors.
Expand All	Expands all categories.
Collapse All	Collapses all categories and clears all selections.
 Filter Advisors	Expands or collapses the Advisor filter. The Advisor filter enables you to filter the Advisors, groups and assets.

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:

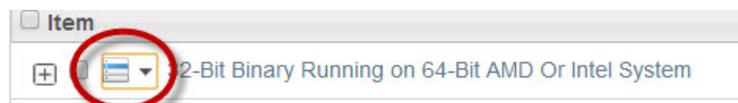
Table 19.2 Advisor Information Listing

Name	Description
Item	Displays the Advisor name, group name, and monitored instance name. To expand the Advisor, click the expand icon.
Info	Click to display a tooltip which describes the Advisor.
Coverage	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 (Covered) .
Schedule	Displays the defined evaluation schedule. If the Advisor is disabled, this field displays Disabled for the level at which it was disabled, Advisor, Group or monitored instance.
Event Handling	Displays the event handling status icons. For more information, see Chapter 18, Event Handling .
Parameters	Displays the Advisor's configuration details, thresholds, and so on.

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 19.3 Advisor Edit Menu Controls

Name	Description
Copy Advisor	<p>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.</p> <div style="display: flex; align-items: center;">  <div> <p>Note</p> <p>This option is only available for expression-based Advisors.</p> </div> </div>
Edit Advisor Configuration	Opens the Edit Advisor dialog. This enables you to change the parameters and schedule of the selected advisor.

Name	Description
Remove Advisor Configuration	Disables the advisor and restores the default values. This is useful if you are experimenting with Advisor configuration, misconfigure the Advisor, and want to start again with the default Advisor configuration.
Disable Advisor	Disables the advisor and its associated graphs.
Delete Advisor	<p>Deletes the selected advisor.</p> <div style="display: flex; align-items: center;">  <div> <p>Note</p> <p>Only available for custom Advisors. It is not possible to delete the default Advisors.</p> </div> </div>

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 same menu is used for each host within the group. 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

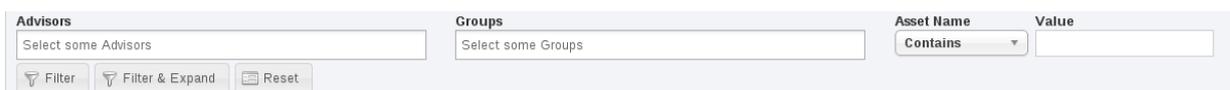


Table 19.4 Advisor Filter Controls

Name	Description
Advisors	Opens a drop-down menu listing all available Advisors. You can select multiple Advisors.

Name	Description
Groups	Opens a drop-down menu listing all defined groups. You can select multiple groups.
Asset Name	Opens a drop-down menu listing the available search types: <ul style="list-style-type: none"> • 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**:

Figure 19.5 Agent Health - General

The screenshot shows a configuration window titled 'General' with the following settings:

- Agent CPU Threshold** (with a help icon):
 - Notice Threshold
 - Warning Threshold
 - Critical Threshold (slider set to 10)
 - Emergency Threshold
- Memory Usage Thresholds (% of max allowed)** (with a help icon):
 - Notice Threshold (slider set to 70)
 - Warning Threshold (slider set to 85)
 - Critical Threshold (slider set to 95)
 - Emergency Threshold
- Moving Average Window (minutes)** (with a help icon):
 - Input: 5
 - Unit: Minutes

At the bottom, there are tabs for 'Communication' and 'Backlog', and 'Save' and 'Cancel' buttons.

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:

Figure 19.6 Threshold Definitions Example

The screenshot shows a configuration window with two tabs: 'Parameters' and 'Schedule'. The 'Parameters' tab is active. Under the 'Thresholds' section, there are four threshold types, each with a checked checkbox and a corresponding numerical value in a text input field:

- Notice Threshold: 75
- Warning Threshold: 85
- Critical Threshold: 95
- Emergency Threshold: 100

At the bottom of the window, there are two buttons: 'Save' (with a green checkmark icon) and 'Cancel' (with a red X icon).

The values shown are taken from the Availability Advisor, **Maximum Connection Limit Nearing or Reached**. The values define the percentage of maximum connections at which an event is logged. For example:

- If the total number of connections is 75-84% of the maximum defined, a **Notice** event is displayed in the **Events** page.
- If the total number of connections is 85-94% of the maximum defined, a **Warning** event is displayed in the **Events** page.
- If the total number of connections is 95-99% of the maximum defined, a **Critical** event is displayed in the **Events** page.
- If the total number of connections is 100% or more of the maximum defined, an **Emergency** event is displayed in the **Events** page.

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](#)

- [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.

Default frequency 06:00:00

Default auto-close enabled yes

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.



Important

The Availability statistics on the main Dashboard Overview page require this advisor to be enabled.

Default auto-close enabled yes

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

Default frequency 00:05:00

Default auto-close enabled no

20.4 Cluster Advisors

This section describes the expression-based Cluster Advisors.

- [Cluster Data Node Data Memory Getting Low](#)
- [Cluster Data Node Has Been Restarted](#)
- [Cluster Data Node Index Memory Getting Low](#)
- [Cluster Data Node Redo Buffer Space Getting Low](#)
- [Cluster Data Node Redo Log Space Getting Low](#)
- [Cluster Data Node Undo Buffer Space Getting Low](#)
- [Cluster Data Node Undo Log Space Getting Low](#)
- [Cluster Data Nodes Not Running](#)
- [Cluster DiskPageBuffer Hit Ratio Is Low](#)
- [Cluster Has Stopped](#)

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

Default auto-close enabled no

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](#)

- [Query Cache Potentially Undersized](#)
- [Table Cache Not Optimal](#)
- [Thread Cache Size May Not Be Optimal](#)

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](#)

- [Thread Cache Not Enabled](#)
- [Thread Pool Stall Limit Too Low](#)
- [Thread Pooling Not Enabled](#)
- [Too Many Concurrent Queries Running](#)

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](#)

- [Slave Master Info/Relay Log Info Not Crash Safe](#)
- [Slave Not Configured As Read Only](#)
- [Slave Not Verifying Checksums When Reading From Relay Log](#)
- [Slave Relay Log Space Is Very Large](#)
- [Slave Relay Logs Not Automatically Purged](#)
- [Slave SQL Processing Not Multi-Threaded](#)
- [Slave SQL Thread Reading From Older Relay Log Than I/O Thread](#)
- [Slave Too Far Behind Master](#)
- [Slave Without REPLICATION SLAVE Accounts](#)

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 and investigate the reasons for the changes.

Default frequency 00:10:00

Default auto-close enabled no

Object Changed: Function Has Been Dropped

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures or functions and investigate the reasons for the changes.

Default frequency 00:10:00

Default auto-close enabled no

Object Changed: Index Has Been Created

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

Default frequency 00:10:00

Default auto-close enabled no

Object Changed: Index Has Been Dropped

For development environments, changes to databases and objects may be a normal occurrence, but not for production environments. It is wise to know when any changes occur in a production environment with respect to any database structures and investigate the reasons for the changes.

Default frequency 00:10:00

Default auto-close enabled no

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](#)

- [Symlinks Are Enabled](#)
- [User Has Rights To Database That Does Not Exist](#)
- [User Has Rights To Table That Does Not Exist](#)
- [Users Can View All Databases On MySQL Server](#)

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.

Default frequency 00:05:00

Default auto-close enabled no

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 01:00:00

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).

Default frequency 06:00:00

Default auto-close enabled no

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).

Default frequency 06:00:00

Default auto-close enabled no

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.

Default frequency 00:05:00

Default auto-close enabled no

Privilege Alterations Detected: Privileges Granted

For development environments, changes to database security privileges may be a normal occurrence, but for production environments it is wise to know when any security changes occur with respect to database privileges, and to ensure that those changes are authorized and required.

Default frequency 00:05:00

Default auto-close enabled no

Privilege Alterations Detected: Privileges Revoked

For development environments, changes to database security privileges may be a normal occurrence, but for production environments it is wise to know when any security changes occur with respect to database privileges, and to ensure that those changes are authorized and required.

Default frequency 00:05:00

Default auto-close enabled no

Root Account Can Login Remotely

By default, MySQL includes a root account with unlimited privileges that is typically used to administer the MySQL server. If possible, accounts with this much power should not allow remote logins in order to limit access to only those users able to login to the machine on which MySQL is running. This helps prevent unauthorized users from accessing and changing the system.

Default frequency 00:05:00

Default auto-close enabled no

Root Account Without Password

The root user account has unlimited privileges and is intended for administrative tasks. Privileged accounts should have strong passwords to prevent unauthorized users from accessing and changing the system.

Default frequency 00:05:00

Default auto-close enabled yes

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	223
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.

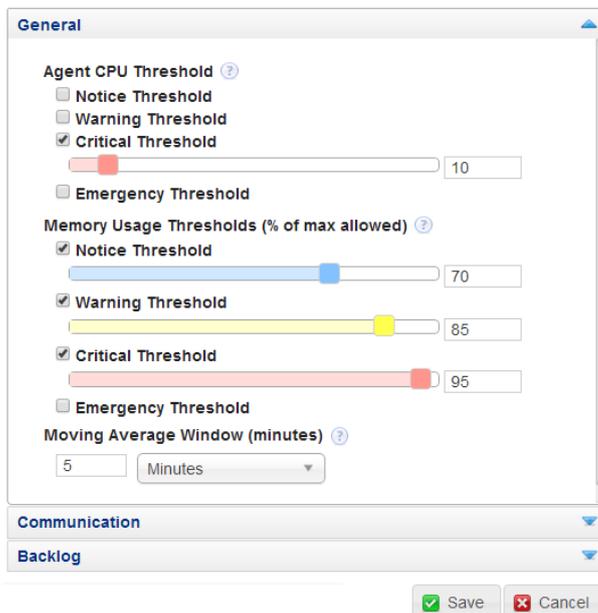
The Agent Health Advisor configuration dialog is divided into the following functional areas:

- [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

The screenshot shows a configuration window titled 'General' with a sub-tab 'Communication'. Under 'Agent Latency Thresholds (minutes)', there are four options: 'Notice Threshold' (unchecked), 'Warning Threshold' (checked, set to 1 minute), 'Critical Threshold' (checked, set to 10 minutes), and 'Emergency Threshold' (unchecked). Under 'HTTP Error Thresholds (% of total requests)', there are four options: 'Notice Threshold' (checked, slider at 10%), 'Warning Threshold' (checked, slider at 20%), 'Critical Threshold' (checked, slider at 30%), and 'Emergency Threshold' (unchecked). At the bottom, there are 'Save' and 'Cancel' buttons.

- **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 exists, 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

The screenshot shows the 'Backlog' configuration window. It has three tabs: 'General', 'Communication', and 'Backlog'. The 'Backlog' tab is active. Under 'Backlog Memory Usage Thresholds (% of max allowed)', there are checkboxes for 'Notice Threshold', 'Warning Threshold', 'Critical Threshold', and 'Emergency Threshold'. The 'Warning Threshold' is checked and set to 80. Under 'Backlog Disk Usage Thresholds (% of max allowed)', there are checkboxes for 'Notice Threshold', 'Warning Threshold', 'Critical Threshold', and 'Emergency Threshold'. The 'Warning Threshold' is checked and set to 80. At the bottom, there are 'Save' and 'Cancel' buttons.

- **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:

- 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 commandline 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

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

Name	Description
Attempt Connection	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.
Alert Level	Level of alert generated if an unmonitored instance is discovered.
Admin User	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 Auto-Create Less Privileged Users 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.
Admin Password	The password for the Admin User.
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. <ul style="list-style-type: none"> • 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.

Name	Description
	If you do not have appropriate users already, they are automatically created if this option is selected. Using these lower privileged users is recommended.
General User	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.
General Password	The password for the user with general privileges.
Limited User	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.
Limited Password	The password for the user with limited privileges.
MySQL Instance Identity Source	Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist. <ul style="list-style-type: none"> • Default: uses either the <code>server_uuid</code> variable, if present, or generates a random new identity. • Host plus Data Directory: uses a hash of the host identity and the path to the MySQL instances data directory to create a unique identity. The <code>host_and_datadir</code> option can only be used when the agent is running on the same host as the MySQL instance for this connection.

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

Figure 21.4 CPU Usage

The screenshot shows the 'CPU Usage' configuration window. It includes a 'Moving Average Window' section with a text input '5' and a dropdown menu set to 'Minutes'. Below this are two sections of thresholds, each with a checked checkbox and a slider control. The 'CPU Usage Thresholds (%)' section has: Notice Threshold (50%), Warning Threshold (75%), Critical Threshold (90%), and Emergency Threshold (disabled). The 'CPU IO Wait Thresholds (%)' section has: Notice Threshold (10%), Warning Threshold (20%), Critical Threshold (25%), and Emergency Threshold (disabled).

- **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.

Figure 21.5 CPU Outliers

The screenshot shows the 'CPU Outliers' configuration window. It contains four settings: 'Enable CPU Outlier Detection' (dropdown menu set to 'No'), 'Minimum Server Count For Outlier Detection' (text input '10'), 'Small Group Notification' (dropdown menu set to 'Yes'), and 'Outlier Percentile' (slider control set to 90%).

- **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.

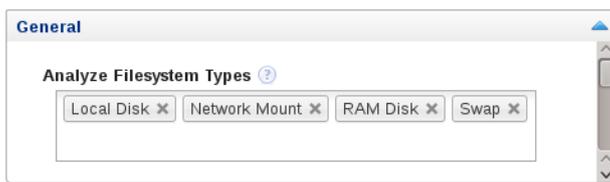
21.6 Filesystem Free Space Advisor

Monitors and graphs the filesystem disk space usage.

Default auto-close enabled yes

General

Figure 21.6 Filesystem - General



The **General** section enables you to choose the filesystem to monitor. The following types are available:

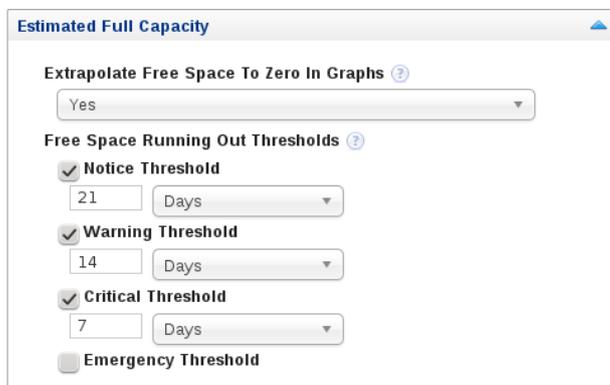
- **Local Disk:** enables monitoring of the local hard disks.
- **Network Mount:** enables monitoring of mounted network filesystems on the monitored server.
- **RAM Disk:** enables monitoring of RAM disks configured on the server.
- **CDROM:** enables monitoring of CD-ROM drives on the server
- **Swap:** 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

Figure 21.7 Filesystem - Estimated Full Capacity



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

Percentage of Space

Free Space Availability Thresholds (% of total space) ?

Notice Threshold

Warning Threshold

Critical Threshold

Emergency Threshold

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

Percentage used in Time Range

Disk Space Consumption Rate Thresholds (% per unit time) ?

Notice Threshold

Warning Threshold

Percent Consumed

Every

Critical Threshold

Emergency Threshold

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

- **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.

Chapter 22 Customizing MySQL Enterprise Monitor

Table of Contents

22.1 Customizing Groups	231
22.2 Creating Advisors and Rules	232
22.2.1 Creating Advisors	232
22.2.2 Overview of Graph Creation	233
22.2.3 Overview of Advisor Creation	234
22.2.4 Variables	235
22.2.5 Thresholds	235
22.2.6 Using Strings	236
22.2.7 Wiki Format	236
22.2.8 Creating a New Advisor: An Example	237
22.2.9 Creating a New Graph: An Example	239
22.3 Custom Data Collection	240
22.3.1 Custom.xml	240
22.3.2 Queries	241
22.3.3 Data Collection Attributes	242
22.4 Event Notification Blackout Periods	244
22.4.1 Scripting Blackouts	245

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.

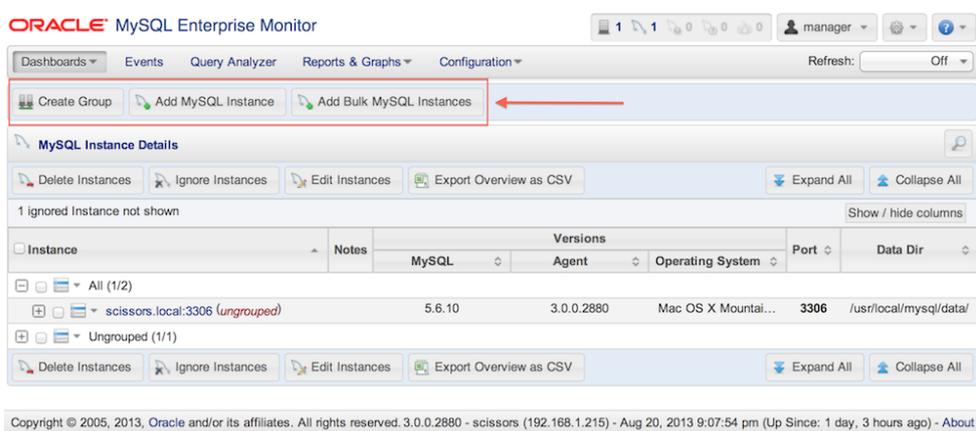


Note

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:

Figure 22.1 Manage Instances example



For more information about configuring groups, see [Section 14.4, "MySQL Instances Dashboard"](#).

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

- 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 dcltem element. The instance defines what data the graph displays, and each dcltem 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`.
- The operators functions listed on this page: <http://www.singularsys.com/jep/doc/html/operators.html>.
- 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.

**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 - ((%Key_reads% / %Key_read_requests%) * 100)
```

If the current value of `%Key_reads%` is 4522 and the current value of `%Key_read_requests%` is 125989, the hit ratio is 96.4%:

```
100 - ((4522 / 125989) * 100)
```

By convention, the Advisors supplied by MySQL use ‘%’ as the delimiter, for example, `%Key_reads%`. This makes variables more readily identifiable.

Variables can be used in the [Description](#), [Advice](#), [Action](#), and [Links](#) attributes of a rule, as well as in expressions. This lets you report the current value of an expression. For instance, you can add the message, “The current value of Key_reads is %Key_reads%.” to the [Advice](#) text box. When this is displayed on the screen, the value of `%Key_reads%` is substituted into the text. If `%Key_reads%` has a value of 4522, the message becomes “The current value of Key_reads is 4522.”

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 - ((%Threads_created% / %Connections%) * 100) < THRESHOLD
```

The `THRESHOLD` is set at 95% for an Info level alert, 85% for a Warning alert, and 75% for a Critical alert, producing alerts of three different levels.

Expressions can be straightforward. The expression for “Binary Logging Not Enabled” (one of the Administration alerts) is:

```
%log_bin% == THRESHOLD
```

When the result is `OFF`, only one alert is triggered: a Warning level alert. You cannot just use the expression `%log_bin% == "OFF"`, because this would not test binary logging against a threshold and so would not result in an alert.

Specify precise conditions when each expression should 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) < 10)
```

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

Example	Description
<code>__bold__</code>	boldface text
<code>~~italic~~</code>	italicize text
<code>\</code>	create a line break
<code>\</code>	create a double line break
<code>\\G</code>	create a backslash
<code>*item 1</code>	create a bulleted list item
<code>#item 1</code>	create a numbered list item
<code>_</code>	use the ‘\’ to escape special characters
<code>{'moreInfo:name url}'</code>	create a hyperlink

So the following Wiki text:

```
Replication is a __very nice feature__ of MySQL.  Replication can be very
useful for solving problems in the following areas:
* Data Distribution
* Load Balancing
* Backup and Recovery
You can check replication status and start a slave using the following
commands: SHOW SLAVE STATUS \\G\\START SLAVE; {'moreInfo:MySQL Manual: Replication
FAQ|http://dev.mysql.com/doc/refman/5.6/en/faqs-replication.html' }
```

Would be translated into the following HTML markup:

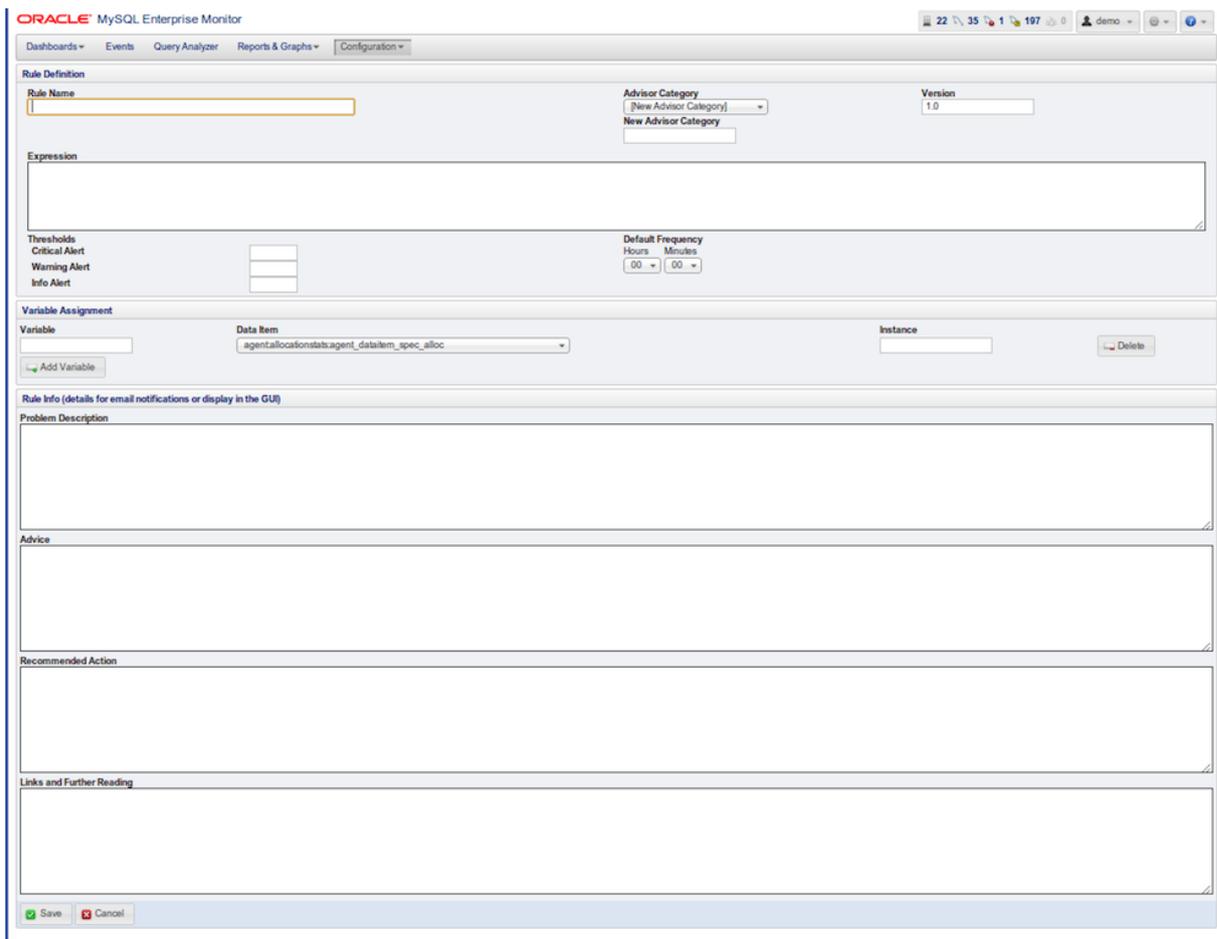
```
Replication is a <b>very nice feature</b> of MySQL.  Replication can be very
useful for solving problems in the following areas:
<ul>
  <li>Data distribution</li>
  <li>Load Balancing</li>
  <li>Backup and recovery</li>
</ul>You can check replication status and start a slave with the following
commands: SHOW SLAVE STATUS \G;<br/>START SLAVE;
<a href="http://dev.mysql.com/doc/refman/5.6/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:

Figure 22.2 Creating a New Advisor



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:

- 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 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.

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 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

Element	Description
<code>classes</code>	Container element for all defined classes.
<code>class</code>	Container element for the definition of the collection.
<code>namespace</code>	Logical grouping for the new data collection item.
<code>classname</code>	Name of the custom data collection. Do not use spaces or special characters in this element.
<code>precondition</code>	(Optional) Query which checks some conditions. If the query returns true, the main query is executed. For example, the precondition query can be

Element	Description
	used to check the version of the MySQL server. See Section 22.3.2.1, “Precondition Queries” for more information.
<code>query</code>	The main query. For more information, see Section 22.3.2.2, “Main Queries”
<code>attributes</code>	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.

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\n* ', '\n',user,'\n@','host,'\n' 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

Name	Description
name	The name of the attribute defined in the AS clause of the query.

Name	Description
counter	Whether the attribute is a counter type. <ul style="list-style-type: none"> <code>true</code>: the attribute is a counter type. <code>false</code>: the attribute is not a counter type.
type	The attribute value type. Possible values are INTEGER, STRING or FLOAT.



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:

```
<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:

```
<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:

```

<class>
  <namespace>mysql</namespace>
  <classname>rpl_semi_sync_vars</classname>
  <query><![CDATA[
SHOW GLOBAL VARIABLES WHERE
Variable_name='rpl_semi_sync_master_timeout' OR
Variable_name='rpl_semi_sync_master_trace_level' OR
Variable_name='rpl_semi_sync_master_wait_no_slave' OR
Variable_name='rpl_semi_sync_master_enabled' OR
Variable_name='rpl_semi_sync_slave_enabled'
]]></query>
<resulttype>CLASS_TYPE_1STCOL_ATTRIBUTES</resulttype>
<attributes>
  <attribute name="rpl_semi_sync_master_timeout" counter="false" type="INTEGER"/>
  <attribute name="rpl_semi_sync_master_trace_level" counter="false" type="INTEGER"/>
  <attribute name="rpl_semi_sync_master_wait_no_slave" counter="false" type="STRING"/>
  <attribute name="rpl_semi_sync_master_enabled" counter="false" type="STRING"/>
  <attribute name="rpl_semi_sync_slave_enabled" counter="false" type="STRING"/>
</attributes>
</class>

```

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`.

```
#!/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:

```
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:

```
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:

```
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 frequency 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 23.1 QRTi value definitions

Type	Default time values	Assigned value	Description	Color
Optimum	100ms	1.00 (100%)	The optimal time frame	Green
Acceptable	4 * Optimum -- 100ms to 400ms	0.50 (50%)	An acceptable time frame	Yellow
Unacceptable	Exceeds Acceptable -- greater than 400ms	0.00 (0%)	An unacceptable time frame	Red

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:

$$((60 + (30 / 2) + (10 * 0)) / 100) = 0.75 .$$

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.

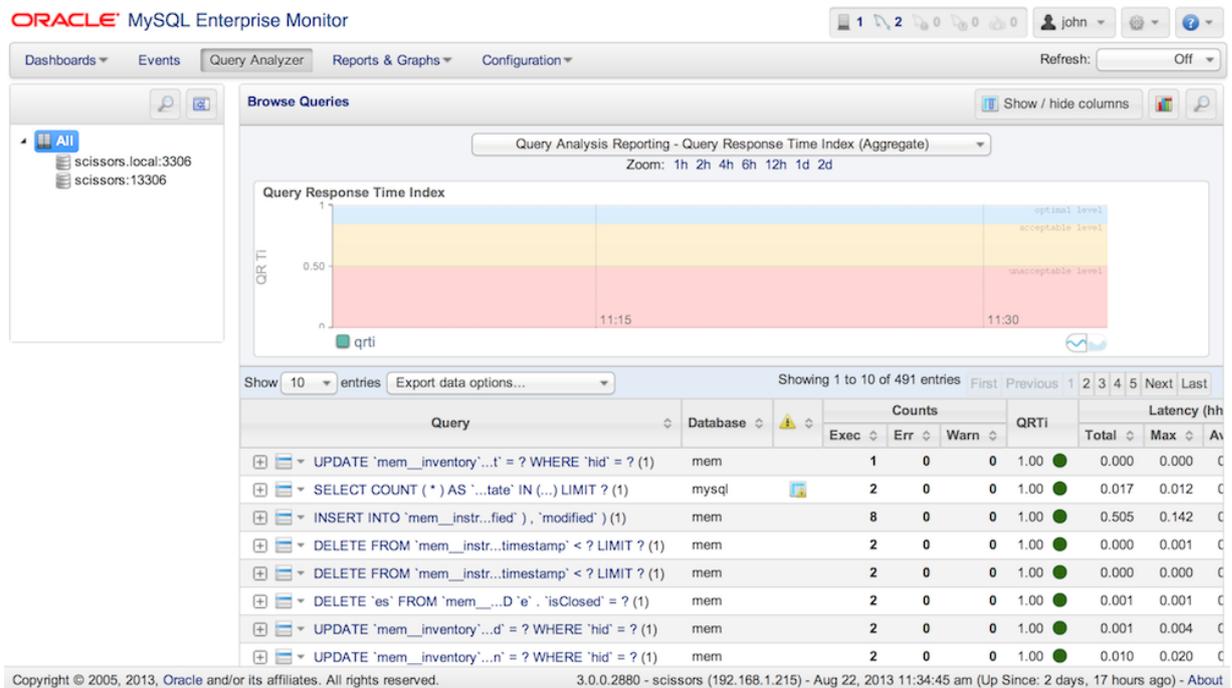
So when doing query optimization, you want to start with the ones that have a QRTi visual pie chart that is 100% red, which means that they also have an actual QRTi value of 0. This means that *all* executions of that query took longer than the acceptable time frame (400ms by default). You can then click 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

Canonical Query | Example Query | Explain Query | Graphs

Overview of information collected and aggregated for queries of this form.

Alias
None specified.

Canonical Form
[truncated](#) | [full](#) | [formatted](#)

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

Execution Time Statistics

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

Row Statistics

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

Execution Summary

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

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

First Seen
Mar 16, 2010 4:37:08 PM

[expand »](#)

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.

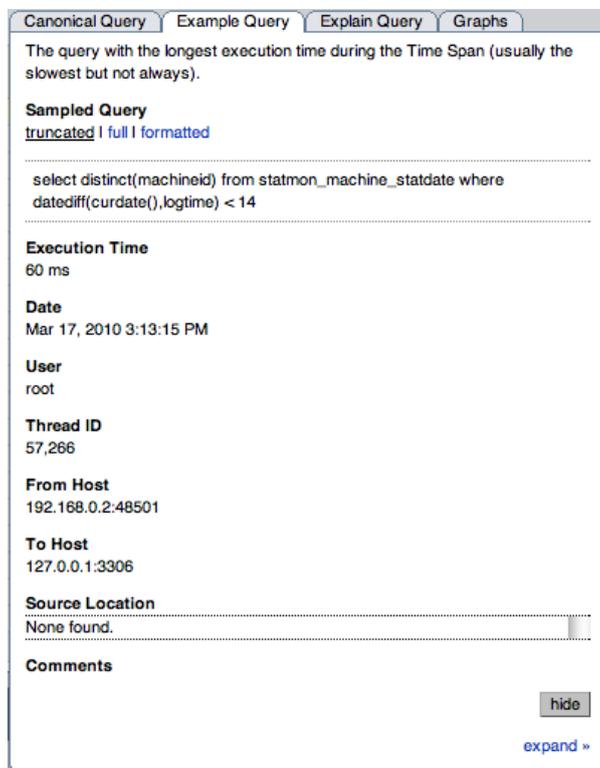
The **Execution Summary** section provides the summary data available in the main table, covering the execution count, and counts of the number of errors, warnings, queries that triggered table scans, and those that indicated a bad index use.

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



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:

Figure 23.4 MySQL Enterprise Monitor User Interface: Explain Query Tab for a Query

Canonical Query									
Example Query									
Explain Query									
Graphs									
Explain of a query that occurred during the Time Span (usually the slowest but not always).									
Explain									
id	select_type	table	type	possible_keys	key	key_len	ref	rows	extra
1	SIMPLE	media_photo_dates	ref	photodates	photodates	2	const	13137	Using where; Using index; Using temporary; Using filesort
1	SIMPLE	media_photo	eq_ref	PRIMARY	PRIMARY	4	intranet_mcsip.media_photo_dates.photoid	1	Using where; Distinct

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



Important

Explain is generated for query data supplied by the MySQL Enterprise Monitor Proxy and Aggregator, Connector/J plugin, and Performance Schema sources.

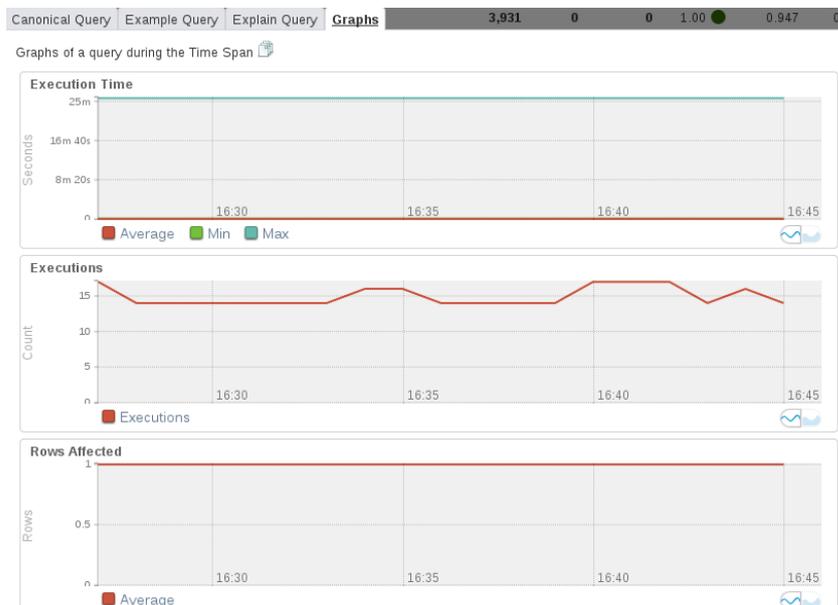
Explain is supported for all DML statements on MySQL 5.6.3 or higher. On earlier versions, only `SELECT` is supported.



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:

Figure 23.5 Graphs Tab for a Query

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

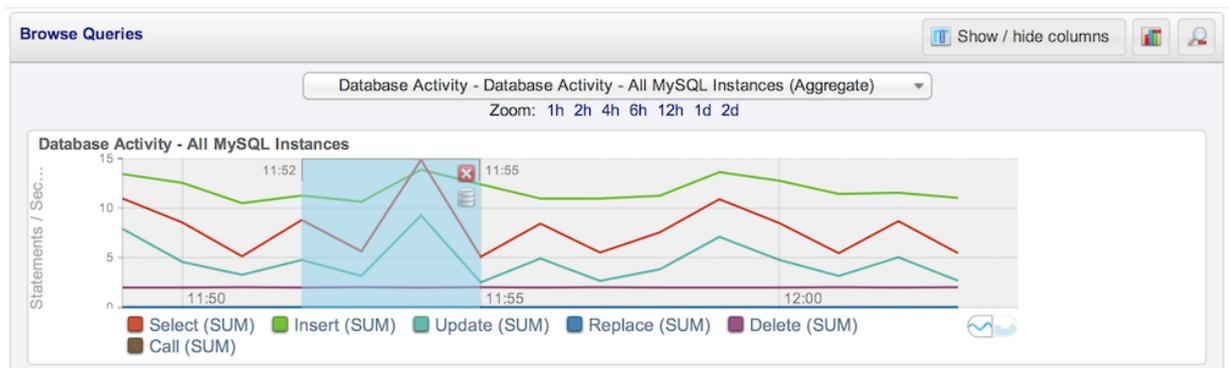
23.3.2 Using Graphs to Identify Queries

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.

Figure 23.7 MySQL Enterprise Monitor User Interface: Query Analyzer Filter Options

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.

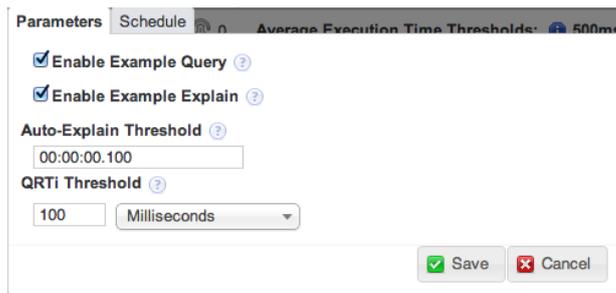
When you have set your filter parameters, you can update the Query Analysis display by clicking the **filter** button. To reset the fields to the default settings click the **reset** button.

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.

These 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 <code>config.properties</code> 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 handle 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 -u service_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 <code>config.properties</code> 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

Component	File Location
Apache/Tomcat	\MySQL\Enterprise\Monitor\apache-tomcat\logs\catalina.out
Repository	\MySQL\Enterprise\Monitor\mysql\data
Configuration Report	\MySQL\Enterprise\Monitor\configuration_report.txt
Service Manager (General)	\MySQL\Enterprise\Monitor\apache-tomcat\logs\mysql-monitor.log
Service Manager (Full / Support)	\MySQL\Enterprise\Monitor\apache-tomcat\logs\mysql-monitor-full.log

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

Operating System	File Location
Windows	C:\Program Files\MySQL\Enterprise\Monitor\support-files\MONITOR.MIB
Unix	/opt/mysql/enterprise/monitor/support-files/MONITOR.MIB
Mac OS X	/Applications/mysql/enterprise/monitor/support-files/MONITOR.MIB

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 D.3 MySQL Enterprise Monitor: Default path of the `config.properties` File

Operating System	File Location
Windows	C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\webapps\ROOT\WEB-INF
Linux and Unix	/opt/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF
Mac OS X	/Applications/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF

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=BMcsacZdrMmM7mrnFExURHDuxp4C3hcrZyxcpc2QhiE\=
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-dbc/configuration.html>.

Table D.4 Optional `config.properties` values

Property Name	Property Type	Default
<code>data_collection_interval</code>	string	00:01:00
<code>dbpool.default.initialSize</code>	integer	20
<code>dbpool.default.maxActive</code>	integer	70
<code>dbpool.default.maxIdle</code>	integer	20

Property Name	Property Type	Default
<code>dbpool.default.maxWaitMillis</code>	string	30 seconds
<code>dbpool.default.minEvictableIdleTimeMillis</code>	string	15 seconds
<code>dbpool.default.minIdle</code>	integer	0
<code>dbpool.default.timeBetweenEvictionRunsMillis</code>	string	5 seconds
<code>dbpool.ui.initialSize</code>	integer	5
<code>dbpool.ui.maxActive</code>	integer	15
<code>dbpool.ui.maxIdle</code>	integer	5
<code>dbpool.ui.maxWaitMillis</code>	string	30 seconds
<code>dbpool.ui.minEvictableIdleTimeMillis</code>	string	15 seconds
<code>dbpool.ui.minIdle</code>	integer	0
<code>dbpool.ui.timeBetweenEvictionRunsMillis</code>	string	5 seconds
<code>internal_perf_enable</code>	boolean	false
<code>internal_perf_server_id</code>	integer	
<code>notify_thread_pool_size</code>	integer	4
<code>quanal.collect</code>	string	00:01:00
<code>supportReport.retention.minutes</code>	string	6 hours
<code>ui.javascript.useClientSideStorage</code>	boolean	false

- `notify_thread_pool_size(4)`

Permitted Values	Type	integer
	Default	4

SMTP and SNMP notifications are sent asynchronously, this controls how many threads are used for this process.

- `thread_pool_size(8)`

Permitted Values	Type	integer
	Default	8

Used to handle background jobs.

- `data_collection_interval(00:01:00)`

Permitted Values	Type	string
	Default	00:01:00

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)`

Permitted Values	Type	boolean
	Default	false

Enables internal performance monitoring for MySQL Enterprise Monitor (requires deploying some graphs from the `resources/` directory).

- `internal_perf_server_id(false)`

Permitted Values	Type	integer
	Default	

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)`

Permitted Values	Type	string
	Default	<code>00:01:00</code>

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)`

Permitted Values	Type	boolean
	Default	<code>false</code>

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)`

Permitted Values	Type	string
	Default	<code>6 hours</code>

The length of time that MySQL Enterprise Monitor will retain the reports generated when using "Support diagnostics" from "Manage Servers".

- `dbpool.ui.initialSize(5)`

Permitted Values	Type	integer
	Default	<code>5</code>

- `dbpool.ui.maxActive(15)`

Permitted Values	Type	integer
	Default	<code>15</code>

- `dbpool.ui.minIdle(0)`

Permitted Values	Type	integer
	Default	<code>0</code>

- `dbpool.ui.maxIdle(5)`

Permitted Values	Type	integer
	Default	<code>5 284</code>

- `dbpool.ui.maxWaitMillis(30 seconds)`

Permitted Values	Type	string
	Default	30 seconds

- `dbpool.ui.timeBetweenEvictionRunsMillis(5 seconds)`

Permitted Values	Type	string
	Default	5 seconds

- `dbpool.ui.minEvictableIdleTimeMillis(15 seconds)`

Permitted Values	Type	string
	Default	15 seconds

- `dbpool.default.initialSize(20)`

Permitted Values	Type	integer
	Default	20

- `dbpool.default.maxActive(70)`

Permitted Values	Type	integer
	Default	70

- `dbpool.default.minIdle(0)`

Permitted Values	Type	integer
	Default	0

- `dbpool.default.maxIdle(5)`

Permitted Values	Type	integer
	Default	20

- `dbpool.default.maxWaitMillis(30 seconds)`

Permitted Values	Type	string
	Default	30 seconds

- `dbpool.default.timeBetweenEvictionRunsMillis(5 seconds)`

Permitted Values	Type	string
	Default	5 seconds

- `dbpool.default.minEvictableIdleTimeMillis(15 seconds)`

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

	Default 15 seconds
--	-----------------------------

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)

Operating System	Path
Microsoft Windows	C:\Program Files\MySQL\Enterprise\Monitor\bin\config.bat
Linux / Solaris	/opt/mysql/enterprise/monitor/bin/config.sh
Mac OS X	/Applications/mysql/enterprise/monitor/bin/config.sh

Use `--help` to view its options:

```
bin/config.sh --help
Usage: java com.mysql.ertools.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/>.

D.2.3 Using The Agent Configuration Script

The `agent.sh` / `agent.bat` script is used to configure an Agent. Its location:

Table D.6 MEM Agent Configuration Tool Location (default)

Operating System	Path
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Agent\bin\agent.bat</code>
Linux / Solaris	<code>/opt/mysql/enterprise/agent/bin/agent.sh</code>
Mac OS X	<code>/Applications/mysql/enterprise/agent/bin/agent.sh</code>

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 E.1 MySQL Enterprise Dashboard: GUI Error Codes and Messages

Error Code	Message
U0001	" <i>replaceable_value</i> " is a required field.
U0002	You must log in to access the requested resource.
U0003	You do not have permissions to access this resource.
U0004	The user field " <i>replaceable_value</i> " must not be null. Either the submission was broken or you have accessed this resource improperly.
U0005	A non-numeric value was submitted for the field " <i>replaceable_value</i> ". Either the submission was broken or you have accessed this resource improperly.
U0007	Unable to connect to download Advisors. <i>replaceable_value</i>
U0009	The uploaded Advisor jar was invalid.
U0010	The username and password combination was not found.
U0011	The user " <i>replaceable_value</i> " does not exist.
U0012	You may not log into the interface with agent credentials.
U0013	At least one user with the "agent" role is required to use the application.
U0014	At least one user with the "manager" role is required to use the application.
U0018	You must provide an admin password.
U0019	The admin passwords do not match.
U0020	You must provide an agent password.
U0021	The agent passwords do not match.
U0023	You must provide a non-zero interval.
U0024	" <i>replaceable_value</i> " is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0025	" <i>replaceable_value</i> " is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0026	Your "To" and "From" dates cannot be the same date.
U0027	You must provide a graph id for display.
U0028	You have exceeded the maximum number of graphs for display. You may only select <i>replaceable_value</i> graphs.
U0029	No graphs ids were found.
U0030	You must provide a non-zero interval.
U0031	Your full-sized width must be greater than or equal to <i>replaceable_value</i> .
U0032	" <i>replaceable_value</i> " is not a valid value for full-sized width.
U0033	Your full-sized height must be greater than or equal to <i>replaceable_value</i> .
U0034	" <i>replaceable_value</i> " is not a valid value for full-sized height.
U0035	Your thumbnail width must be greater than or equal to <i>replaceable_value</i> .
U0036	" <i>replaceable_value</i> " is not a valid value for thumbnail width.
U0037	Your thumbnail height must be greater than or equal to <i>replaceable_value</i> .
U0038	" <i>replaceable_value</i> " is not a valid value for thumbnail height.
U0039	You must select one or more events to close.

Error Code	Message
U0041	" <i>replaceable_value</i> " is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0042	" <i>replaceable_value</i> " is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0043	You must provide a non-zero interval.
U0044	Your width must be greater than or equal to <i>replaceable_value</i> .
U0045	" <i>replaceable_value</i> " is not a valid value for width.
U0046	Your height must be greater than or equal to <i>replaceable_value</i> .
U0047	" <i>replaceable_value</i> " is not a valid value for height.
U0048	You must provide a non-zero interval.
U0049	" <i>replaceable_value</i> " is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0050	" <i>replaceable_value</i> " is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0051	Your "To" and "From" dates cannot be the same date.
U0052	Your width must be greater than or equal to <i>replaceable_value</i> .
U0053	" <i>replaceable_value</i> " is not a valid value for width.
U0054	Your height must be greater than or equal to <i>replaceable_value</i> .
U0055	" <i>replaceable_value</i> " is not a valid value for height.
U0056	You must select one or more advisors.
U0057	No schedule was found with id " <i>replaceable_value</i> ".
U0058	You must select one or more rules to schedule.
U0059	There was a problem scheduling " <i>replaceable_value</i> " against " <i>replaceable_value</i> ".
U0060	No rule was found with the uuid " <i>replaceable_value</i> ".
U0061	Advisor name must not be empty.
U0062	Advisor " <i>replaceable_value</i> " already exists.
U0063	New advisor name must not be empty.
U0064	Advisor id must not be null.
U0065	No advisor was found with the id <i>replaceable_value</i> .
U0066	Rule uuid must not be null.
U0067	No rule was found with the uuid " <i>replaceable_value</i> ".
U0068	One or more variables is required to make a functioning rule.
U0069	Each defined variable must be mapped to an instance. The variable " <i>replaceable_value</i> " is missing the instance to collect against.
U0070	Please choose an existing Advisor or create a new one with a non-empty name.
U0071	The default frequency must be a non-zero value.
U0072	You must enter a rule name.
U0075	You must specify a file for upload.
U0077	You must provide an email address.
U0078	The provided email address appears to be invalid.
U0079	A valid SMTP server address must be given.

Error Code	Message
U0080	Your passwords do not match.
U0081	A valid SNMP target must be given.
U0082	A valid SNMP port must be given.
U0083	A valid SNMP community string must be given.
U0084	The given file could not be found.
U0086	An invalid file name was given.
U0088	An invalid SNMP severity level was given.
U0090	An invalid data life span was given.
U0091	An invalid re-inventory frequency was given.
U0093	The user name is missing.
U0095	An invalid time zone was given.
U0096	An invalid locale was given.
U0097	The group name is invalid.
U0098	The recipients list is empty.
U0099	Cannot find the e-mail target.
U0100	The given target id is invalid.
U0101	The group name must not be empty.
U0102	The server name must not be empty.
U0103	An invalid group id was given.
U0104	The group already exists.
U0105	This group name is already in use. Enter a different name.
U0106	Cannot find group with id <i>replaceable_value</i> .
U0107	Cannot find server with id <i>replaceable_value</i> .
U0108	An invalid server id was given.
U0109	Invalid Username
U0110	Invalid user role specified.
U0111	A user with the given name already exists.
U0112	The password must not be empty.
U0113	The user " <i>replaceable_value</i> " does not exist.
U0114	The log name must not be empty.
U0115	An invalid log level was given.
U0117	An invalid filter was set.
U0118	" <i>replaceable_value</i> " is an invalid "From" date. Dates are in the format YYYY-MM-DD.
U0119	" <i>replaceable_value</i> " is an invalid "To" date. Dates are in the format YYYY-MM-DD.
U0120	No event was found with the id " <i>replaceable_value</i> ".
U0121	No category was found with the id " <i>replaceable_value</i> ".
U0122	You must select one or more instances to add.
U0123	No log was found with the name " <i>replaceable_value</i> ".
U0124	No schedule was found with the id " <i>replaceable_value</i> ".

Error Code	Message
U0125	No notification group was found with the id " <i>replaceable_value</i> ".
U0126	Unable to retrieve Advisors at this time.
U0128	You must specify a file for upload.
U0129	The uploaded Graph XML was invalid.
U0130	The uploaded Graph had an insufficient version number, and was not loaded.
U0131	No group was found with the id " <i>replaceable_value</i> ".
U0132	No user was found with the name " <i>replaceable_value</i> ".
U0133	No log was found with the key " <i>replaceable_value</i> ".
U0134	An invalid e-mail address was given.
U0135	No user was found with the id " <i>replaceable_value</i> ".
U0136	No graph was found with the id " <i>replaceable_value</i> ".
U0137	You must provide a query summary id for display.
U0138	No query summary was found with the id " <i>replaceable_value</i> ".
U0139	" <i>replaceable_value</i> " is not a valid UUID.
U0140	You are not authorized to access this resource.
U0141	No graph was found with the uuid " <i>replaceable_value</i> ".
U0142	The target name is already in use.
U0143	No key was provided to sort on.
U0145	U0120 Unable to import Advisor Jar. You may download the jar manually from the My Oracle Support website and import it on this page.
U0146	Unable to schedule rule " <i>replaceable_value</i> " due to " <i>replaceable_value.replaceable_value</i> " data not being collected from server " <i>replaceable_value</i> ". It may be an unsupported collection for that server.
U0147	The requested feature, <i>replaceable_value</i> , has not been implemented.
U0148	You must select one or more servers.
U0149	You must provide a non-empty alias.
U0150	No instance was found with the id " <i>replaceable_value</i> ".
U0151	Instance already exists with name " <i>replaceable_value</i> ".
U0152	No instance name provided
U0154	Query Analyzer Data purge value cannot be greater than Historic Data purge value.
U0156	The database cannot parse the regular expression: <i>replaceable_value</i> .
U0157	Query Explain support requires Query Example capture to be enabled.
U0162	The admin and agent usernames must be different.
U0163	The requested alias is too long; aliases must be less than <i>replaceable_value</i> characters.
U0164	" <i>replaceable_value</i> " is not a DNS resolvable hostname or IP address.
U0165	Heat chart rule " <i>replaceable_value</i> " scheduled against " <i>replaceable_value</i> " may not be unscheduled. This rule is required for the product to function properly.
U0166	Heat chart rule " <i>replaceable_value</i> " scheduled against " <i>replaceable_value</i> " may not be disabled. This rule is required for the product to function properly.
U0167	Bad graph identifier.
U0168	There is no graph with the given identifier.

Error Code	Message
U0169	Graph width " <i>replaceable_value</i> " is invalid.
U0170	Graph height " <i>replaceable_value</i> " is invalid.
U0171	Graph width " <i>replaceable_value</i> " is below minimum " <i>replaceable_value</i> ".
U0172	Graph height " <i>replaceable_value</i> " is below minimum " <i>replaceable_value</i> ".
U0173	Interval " <i>replaceable_value</i> " is not valid.
U0175	A HTTP proxy host is required; the port is optional.
U0176	Unable to parse the HTTP host " <i>replaceable_value</i> ".
U0177	Unable to parse the HTTP proxy port number " <i>replaceable_value</i> ".
U0178	An invalid hostname or port was given.
U0179	The HTTP proxy passwords must match.
U0180	You cannot change the role of the logged-in user.
U0181	No updates are available to install at this time.
U0182	There was an error formatting the Advisor Readme.
U0183	You must provide a primary hostname.
U0186	You must provide a login when not using anonymous binds.
U0187	You must provide a password attribute when using comparison mode.
U0188	You must provide a password digest when using comparison mode.
U0189	Unknown password digest provided.
U0191	User or role pattern required when using role mapping.
U0192	You must provide a role search pattern attribute.
U0193	You must provide at least one mapping for the "admin" role.
U0194	You must provide at least one mapping for the "dba" role.
U0195	You must provide at least one mapping for the "agent" role.
U0201	You must provide at least one mapping for the "read-only" role.
U0202	<i>replaceable_value</i> heat chart {0,choice,1#rule 1<rules} against <i>replaceable_value</i> {1,choice,1#server 1<servers} will not be unscheduled. {0,choice,1#This rule is 1<These rules are} required for the product to function properly.
U0203	No schedules were found for the rule.
U0204	Delete without confirmation disallowed.
U0205	Nothing is selected.
U0206	No file was provided.
U0207	Unable to determine the type of this file.
U0208	Nothing is selected.
U0209	The selection is invalid.
U0210	A version (e.g., "1.0") is required.
U0211	<i>replaceable_value</i> heat chart {0,choice,1#rule 1<rules} against <i>replaceable_value</i> {1,choice,1#server 1<servers} will not be disabled. {0,choice,1#This rule is 1<These rules are} required for the product to function properly.
U0212	You must specify user search criteria.
U0213	You must choose user search by pattern or by criteria.

Error Code	Message
U0214	The uploaded Advisor jar is a Custom Rule/Graph Export Bundle, and must be uploaded to the Import/Export page.
U0215	Changing the frequency for this rule is not supported.
U0216	Notes provided were too long (limit 10000 characters)
U0217	No notes provided
U0218	No servers were selected, please select the servers you want to modify and try again.
U0219	The interval " <i>replaceable_value</i> " could not be parsed. Intervals must be in hh:mm:ss <i>replaceable_value</i> sec format.
U0220	You must provide at least one substitution expression when selecting the transform with substitution expression option.
U0221	Invalid entry in substitution expression field " <i>replaceable_value</i> ". Entry is not a well formed substitution expression.
U0222	Invalid entry in substitution expression field " <i>replaceable_value</i> ". The substitution pattern " <i>replaceable_value</i> " is not valid.
U0223	Invalid entry in substitution expression field " <i>replaceable_value</i> ". The substitution text " <i>replaceable_value</i> " is not valid.
U0315	Script content file does not exist.
U0316	Script runtime does not exist.
U0317	No event handler was found with the id " <i>replaceable_value</i> ".

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

Error Code	Message
E0001	Internal Error: <i>replaceable_value</i>
E0100	Invalid user role " <i>replaceable_value</i> ". Only "manager", "dba", and "agent" are valid.
E0101	Invalid user name " <i>replaceable_value</i> ". User names must be alphanumeric and between 1-32 characters in length.
E0102	Invalid user password. Passwords must contain at least one number.
E0103	Cannot delete manager user.
E0104	User cannot change role.
E0105	User " <i>replaceable_value</i> " already exists.
E0106	User " <i>replaceable_value</i> " does not exist.
E0107	User role " <i>replaceable_value</i> " is not authorized.
E0108	User " <i>replaceable_value</i> " has been previously deleted and is disabled.
E0200	No mapping exists to map the user " <i>replaceable_value</i> " to a MySQL Network user.
E0201	A mapping already exists for the user " <i>replaceable_value</i> " to a MySQL Network user.
E0202	MySQL Network is currently unavailable or could not be reached.
E0203	MySQL Network reported no alerts.
E0204	MySQL Network Advisor returned no data. Reason: " <i>replaceable_value</i> ".
E0205	Your Advisors are currently up to date.

Error Code	Message
E0206	Your MySQL Enterprise trial expired after " <i>replaceable_value</i> ".
E0207	The MySQL Enterprise subscription key is not valid.
E0208	No MySQL Enterprise subscription information is installed.
E0209	The MySQL Enterprise subscription key has expired.
E0210	User: <i>replaceable_value</i> Requires role: <i>replaceable_value</i>
E0211	You do not have permissions to access this resource.
E0212	The import of a duplicate MySQL Enterprise subscription trial key was rejected.
E0213	Unable to decode key from file: <i>replaceable_value</i> .
E0300	Can not delete default group.
E0301	Group " <i>replaceable_value</i> " already exists.
E0302	Group " <i>replaceable_value</i> " does not exist.
E0303	Invalid group name " <i>replaceable_value</i> ". Group name must be alphanumeric and less than 128 characters.
E0304	Group element " <i>replaceable_value</i> " already exists.
E0305	Group element " <i>replaceable_value</i> " does not exist in the group " <i>replaceable_value</i> ".
E0400	Agent " <i>replaceable_value</i> " does not exist.
E0401	Agent payload parameter NULL.
E0402	Agent task " <i>replaceable_value</i> " does not exist.
E0403	There has been a problem de-serializing this agent's request. Please ensure all agents are version <i>replaceable_value</i> or greater.
E0500	Server " <i>replaceable_value</i> " already exists.
E0501	Server " <i>replaceable_value</i> " does not exist.
E0502	Server " <i>replaceable_value</i> " is actively being monitored and cannot be deleted. The last agent contact for this server was on <i>replaceable_value</i> .
E0503	Server " <i>replaceable_value</i> " is being migrated and cannot be deleted. Please halt the active migration first.
E0600	Database " <i>replaceable_value</i> " already exists.
E0601	Database " <i>replaceable_value</i> " does not exist.
E0700	Threshold " <i>replaceable_value</i> " already exists.
E0701	Threshold " <i>replaceable_value</i> " does not exist.
E0702	Data collection item does not exist.
E0703	serverName or groupName have to be set.
E0800	Data collection item does not exist.
E0801	Data collection is already occurring for " <i>replaceable_value</i> ".
E0802	Data collection does not exist for " <i>replaceable_value</i> ".
E0803	Invalid lifespan. Data life must be greater than 0 days.
E0804	Invalid namespace. Supported namespaces include "mysql" and "os". <i>replaceable_value</i>
E0805	Invalid instance. The instance must be a valid for data collection.
E0806	Invalid source. The source must be valid for data collection.
E0807	Invalid attribute. The attrib must be valid for data collection.

Error Code	Message
E0808	Invalid frequency: " <i>replaceable_value</i> ". The frequency must be at least 1 minute and be in the format HH:MM.
E0809	Invalid port. If the URI uses the MYSQL namespace it must include a server port.
E0810	Invalid server. The server " <i>replaceable_value</i> " must exist in the current inventory.
E0811	Invalid server. For an OS namespace, the server must not include the MySQL server port.
E0812	A SQL failure occurred while saving the data collection schedule. " <i>replaceable_value</i> " Please see the error logs for details.
E0813	Invalid server id " <i>replaceable_value</i> ".
E0814	No data items exist for server " <i>replaceable_value</i> ".
E0815	Value " <i>replaceable_value</i> " not valid for " <i>replaceable_value</i> ".
E0816	categoryName has to be set
E0817	You must specify either a server or group in the data collection schedule.
E0818	The server " <i>replaceable_value</i> " does not support item(s) " <i>replaceable_value</i> " required by monitor " <i>replaceable_value</i> ".
E0819	Invalid data type: " <i>replaceable_value</i> ".
E0900	Advisor id " <i>replaceable_value</i> " does not exist.
E0901	Advisor name " <i>replaceable_value</i> " does not exist.
E0902	Advisor XML is invalid.
E0903	Category " <i>replaceable_value</i> " does not exist.
E0904	Invalid server list.
E0905	SKIPPED " <i>replaceable_value</i> " for <i>replaceable_value</i> because Agent for <i>replaceable_value</i> is down.
E0906	SKIPPED " <i>replaceable_value</i> " for <i>replaceable_value</i> because mysqld on <i>replaceable_value</i> is down.
E0907	Category " <i>replaceable_value</i> " cannot be modified.
E0908	Category " <i>replaceable_value</i> " cannot be deleted.
E0909	Category " <i>replaceable_value</i> " already exists.
E0910	Advisor bundle version <i>replaceable_value</i> is incompatible with server version <i>replaceable_value</i> .
E0911	Heat chart rule " <i>replaceable_value</i> " scheduled against " <i>replaceable_value</i> " may not be unscheduled.
E0912	Heat chart rule " <i>replaceable_value</i> " scheduled against " <i>replaceable_value</i> " may not be disabled.
E1100	Graph time format " <i>replaceable_value</i> " is invalid. Must be in the format HH:MM:SS.
E1101	Graph size height and width must be specified.
E1102	Graph " <i>replaceable_value</i> " does not exist.
E1103	Graph " <i>replaceable_value</i> " already exists.
E1104	Graph " <i>replaceable_value</i> " contains a data collection item that can not be identified.
E1200	Failed to retrieve identity column.
E1201	Required parameter " <i>replaceable_value</i> " was NULL.

Error Code	Message
E1202	Only SELECT and SHOW commands are allowed via this interface.
E1203	Invalid timezone " <i>replaceable_value</i> ".
E1204	Invalid interval " <i>replaceable_value</i> ".
E1205	Could not find object " <i>replaceable_value</i> " in cache " <i>replaceable_value</i> ".
E1206	Feature is not implemented. Parameters: <i>replaceable_value</i>
E1300	Advisor " <i>replaceable_value</i> " already exists.
E1301	Advisor " <i>replaceable_value</i> " does not exist.
E1302	Schedule does not exist. " <i>replaceable_value</i> " " <i>replaceable_value</i> "
E1303	JEP error: expression : " <i>replaceable_value</i> ", message: " <i>replaceable_value</i> ".
E1304	Advisor " <i>replaceable_value</i> " contains a data collection item that can not be identified.
E1305	Alarm level " <i>replaceable_value</i> " is not valid.
E1306	Schedule already exists. " <i>replaceable_value</i> " " <i>replaceable_value</i> ".
E1307	Advisor " <i>replaceable_value</i> " is a read only MySQL Network certified Advisor. Please copy the rule before editing.
E1308	Advisor " <i>replaceable_value</i> " is currently scheduled against one or more monitored MySQL servers and cannot be removed.
E1309	Could not render text. " <i>replaceable_value</i> ".
E1310	No open event for " <i>replaceable_value</i> " on server " <i>replaceable_value</i> ". Perhaps already closed.
E1311	Alarm with id " <i>replaceable_value</i> " does not exist.
E1400	Invalid missed agent heartbeat threshold value.
E1401	Notification entry " <i>replaceable_value</i> " does not exist.
E1402	Agent " <i>replaceable_value</i> " is using session " <i>replaceable_value</i> " but session " <i>replaceable_value</i> " was requested.
E1403	Server name " <i>replaceable_value</i> " is in use by another agent with uuid " <i>replaceable_value</i> ".
E1500	Notification target " <i>replaceable_value</i> " does not exist.
E1501	Invalid email address " <i>replaceable_value</i> ".
E1502	Notification target " <i>replaceable_value</i> " already exists.
E1503	Invalid notification target name " <i>replaceable_value</i> ".
E1504	Invalid monitor name, it must not be empty.
E1505	No variable assignments given, you must define at least one.
E1506	SMTP authentication failed.
E1507	SMTP send failed
E1508	Invalid SNMP target " <i>replaceable_value</i> ".
E1509	Invalid SNMP trap type " <i>replaceable_value</i> ".
E1600	Log " <i>replaceable_value</i> " does not exist.
E1700	Authentication Mechanism was null
E1701	Unsupported Authentication Mechanism <i>replaceable_value</i>
E1702	Incorrect password for user <i>replaceable_value</i>
E1800	Invalid data type: " <i>replaceable_value</i> ", new value: " <i>replaceable_value</i> ".

Error Code	Message
E1801	Invalid value: " <i>replaceable_value</i> " for data type " <i>replaceable_value</i> " for item " <i>replaceable_value</i> ".
E1802	Invalid expression: " <i>replaceable_value</i> ", exception: " <i>replaceable_value</i> ", raw expression: " <i>replaceable_value</i> ". substitute values: " <i>replaceable_value</i> ".
E1900	History not found id: " <i>replaceable_value</i> ", schedule: " <i>replaceable_value</i> ", expression: " <i>replaceable_value</i> ".
E2000	Tag not found, id: " <i>replaceable_value</i> ".
E2101	Unable to collect " <i>replaceable_value</i> " on " <i>replaceable_value</i> " for instance " <i>replaceable_value</i> ".
E9000	<i>replaceable_value</i>
E9001	<i>replaceable_value</i> from: <i>replaceable_value</i>

Appendix 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

Namespace	Namespace Type	Attribute	Type	Description
agent	allocationstats	agent_dataitem_spec_alloc	long_counter	
agent	allocationstats	agent_dataitem_spec_free	long_counter	
agent	allocationstats	agent_item_attribute_alloc	long_counter	
agent	allocationstats	agent_item_attribute_free	long_counter	
agent	allocationstats	agent_item_class_alloc	long_counter	
agent	allocationstats	agent_item_class_config_alloc	long_counter	
agent	allocationstats	agent_item_class_config_free	long_counter	
agent	allocationstats	agent_item_class_free	long_counter	
agent	allocationstats	agent_item_classes_alloc	long_counter	
agent	allocationstats	agent_item_classes_free	long_counter	
agent	allocationstats	agent_item_instance_alloc	long_counter	
agent	allocationstats	agent_item_instance_free	long_counter	
agent	allocationstats	agent_item_instance_name_alloc	long_counter	
agent	allocationstats	agent_item_instance_name_free	long_counter	
agent	allocationstats	agent_item_namespaces_alloc	long_counter	
agent	allocationstats	agent_item_namespaces_free	long_counter	
agent	allocationstats	agent_item_value_alloc	long_counter	
agent	allocationstats	agent_item_value_free	long_counter	
agent	allocationstats	agent_mysqld_alloc	long_counter	
agent	allocationstats	agent_mysqld_free	long_counter	
agent	allocationstats	agent_plugin_alloc	long_counter	
agent	allocationstats	agent_plugin_free	long_counter	
agent	allocationstats	agent_target_alloc	long_counter	
agent	allocationstats	agent_target_free	long_counter	
agent	allocationstats	agent_task_alloc	long_counter	
agent	allocationstats	agent_task_free	long_counter	
agent	allocationstats	curl_mem_alloc	long_counter	
agent	allocationstats	curl_mem_bytes	long	
agent	allocationstats	curl_mem_bytes_max	long	
agent	allocationstats	curl_mem_free	long_counter	
agent	allocationstats	job_response_alloc	long_counter	
agent	allocationstats	job_response_data_alloc	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
agent	allocationstats	job_response_data_free	long_counter	
agent	allocationstats	job_response_datum_alloc	long_counter	
agent	allocationstats	job_response_datum_free	long_counter	
agent	allocationstats	job_response_free	long_counter	
agent	allocationstats	job_response_string_alloc	long_counter	
agent	allocationstats	job_response_string_free	long_counter	
agent	allocationstats	job_task_alloc	long_counter	
agent	allocationstats	job_task_free	long_counter	
agent	allocationstats	network_backlog_alloc	long_counter	
agent	allocationstats	network_backlog_free	long_counter	
agent	allocationstats	network_backlog_node_alloc	long_counter	
agent	allocationstats	network_backlog_node_bytes	long	
agent	allocationstats	network_backlog_node_free	long_counter	
agent	allocationstats	scheduler_job_alloc	long_counter	
agent	allocationstats	scheduler_job_free	long_counter	
agent	allocationstats	xml_mem_alloc	long_counter	
agent	allocationstats	xml_mem_bytes	long	
agent	allocationstats	xml_mem_bytes_max	long	
agent	allocationstats	xml_mem_free	long_counter	
agent	chassisstats	glib_mem_alloc	long_counter	
agent	chassisstats	glib_mem_bytes	long	
agent	chassisstats	glib_mem_bytes_max	long	
agent	chassisstats	glib_mem_free	long_counter	
agent	chassisstats	lua_mem_alloc	long_counter	
agent	chassisstats	lua_mem_bytes	long	
agent	chassisstats	lua_mem_bytes_max	long	
agent	chassisstats	lua_mem_free	long_counter	
agent	lua	mem_size	long	
agent	proc	egid	long	
agent	proc	eid	long	
agent	proc	gid	long	
agent	proc	mem_major_faults	long_counter	
agent	proc	mem_minor_faults	long_counter	
agent	proc	mem_page_faults	long_counter	
agent	proc	mem_resident	long	
agent	proc	mem_share	long	
agent	proc	mem_size	long	
agent	proc	name	string	
agent	proc	nice	long	
agent	proc	ppid	long	
agent	proc	priority	long	

Namespace	Namespace Type	Attribute	Type	Description
agent	proc	processor	long	
agent	proc	state	long	
agent	proc	threads	long	
agent	proc	time_start_time	long	
agent	proc	time_sys	long_counter	
agent	proc	time_total	long_counter	
agent	proc	time_user	long_counter	
agent	proc	tty	long	
agent	proc	uid	long	
monitor	resourceBundle	subscription	string	
monitor	resourceBundle	version	string	
mysql	account_old_password	password	string	
mysql	active_count	active_count	long	
mysql	Agent	agent.latency	long	
mysql	Agent	agent.reachable	string	
mysql	Agent	host_id	string	
mysql	Agent	name	string	
mysql	Agent	quanDataOutOfBand	string	
mysql	Agent	version	string	
mysql	anonymous_user	user_count	long	
mysql	auto_increment_limit	table_list	string	
mysql	avail_count	avail_count	long	
mysql	broad_host_specifier	user	string	A list of users whose host in the mysql.user table meets the condition: WHERE host = '%'; a single string in wiki markup format.
mysql	cluster_data_node	data_memory_total	long	
mysql	cluster_data_node	data_memory_used	long	
mysql	cluster_data_node	index_memory_total	long	
mysql	cluster_data_node	index_memory_used	long	
mysql	cluster_data_node	redo_buffer_total	long	
mysql	cluster_data_node	redo_buffer_used	long	
mysql	cluster_data_node	redo_logspace_total	long	
mysql	cluster_data_node	redo_logspace_used	long	
mysql	cluster_data_node	undo_buffer_total	long	
mysql	cluster_data_node	undo_buffer_used	long	
mysql	cluster_data_node	undo_logspace_total	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	cluster_data_node	undo_logspace	long	
mysql	cluster_data_nodes	not_started	string	
mysql	column	Default	string	The default value of the column.
mysql	column	Extra	string	Any additional information about the column.
mysql	column	Field	string	The name of the column.
mysql	column	Key	string	Whether the column is indexed.
mysql	column	Null	string	Whether NULL is allowed.
mysql	column	Type	string	The data type of the column.
mysql	committed_count	committed_count	long	
mysql	Database	Database	string	
mysql	Database	name	string	
mysql	Explain	extra	string	
mysql	Explain	id	long	
mysql	Explain	key	string	
mysql	Explain	key_len	string	
mysql	Explain	possible_keys	string	
mysql	Explain	ref	string	
mysql	Explain	rows	long	
mysql	Explain	select_type	string	
mysql	Explain	table	string	
mysql	Explain	type	string	
mysql	falcon_database_io	logical_reads	long	
mysql	falcon_database_io	physical_reads	long	
mysql	falcon_record_cache_summary	recordcache_mb	double	
mysql	falcon_record_cache_summary	space_mb	double	
mysql	falcon_system_memory_summary	mem_mb	double	
mysql	falcon_system_memory_summary	total_mem_mb	double	
mysql	falcon_transaction_summary	committed_txns	long	
mysql	falcon_transaction_summary	back_txns	long	
mysql	global_privileges	user	string	
mysql	grant_privileges	user_spec	string	
mysql	inappropriate_slave_privileges	privileges	string	

Namespace	Namespace Type	Attribute	Type	Description
mysql	index	Cardinality	long	An estimate of the number of unique values (cardinality) in the index.
mysql	index	Collation	string	How the column is sorted in the index.
mysql	index	Column_name	string	The column name.
mysql	index	Comment	string	Remarks about the index.
mysql	index	Index_type	string	The index method used (BTREE, FULLTEXT, HASH, RTREE)
mysql	index	Key_name	string	The name of the index.
mysql	index	Non_unique	long	Whether the index can contain duplicates.
mysql	index	Null	string	Whether the column may contain NULL values.
mysql	index	Packed	string	Whether the key is packed.
mysql	index	Seq_in_index	long	The column sequence number in the index, starting with 1
mysql	index	Sub_part	string	The number of indexed characters if the column is only partly indexed.
mysql	index	Table	string	The name of the table associated with the index.
mysql	innodb_active_transactions	actions	long	
mysql	innodb_blocked_transactions	actions	long	
mysql	innodb_compression_time	compress_seconds	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodb_compression	time_compress_seconds	long_counter	
mysql	innodb_lock_wait_transactions		long	
mysql	innodb_plugins	innodb_plugin_list	string	
mysql	innodbstatus	innodb_bp_add_alloc	long	The total memory allocated for the additional buffer pool measured in bytes.
mysql	innodbstatus	innodb_bp_created_per_sec	double	The number of buffer pool pages created per second.
mysql	innodbstatus	innodb_bp_db_pages	long	The current number of buffer pool pages.
mysql	innodbstatus	innodb_bp_dictionary_alloc	long	Size in bytes allocated to the data dictionary tables/indexes
mysql	innodbstatus	innodb_bp_free_buffers	long	The current number of free buffer pool pages.
mysql	innodbstatus	innodb_bp_hit_rate	long	The buffer pool hit rate.
mysql	innodbstatus	innodb_bp_io_cur_pages	long	The number of Buffer Pool LRU list pages accessed total, for the last second
mysql	innodbstatus	innodb_bp_io_sum_pages	long	The number of Buffer Pool LRU list pages accessed total, for the last 50 seconds
mysql	innodbstatus	innodb_bp_io_unzip_cur_pages	long	The number of Buffer Pool unzip_LRU list pages accessed total, for the last second

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodbstatus	<code>innodb_bp_io_unzip_sum_pages</code>	long	The number of Buffer Pool unzip_LRU list pages accessed total, for the last 50 seconds
mysql	innodbstatus	<code>innodb_bp_lru_len</code>	long	The size in 16Kb pages of the Buffer Pool LRU list
mysql	innodbstatus	<code>innodb_bp_modified_pages</code>	long	The current number of pages modified.
mysql	innodbstatus	<code>innodb_bp_not_young_hit_rate</code>	long	The rate at which pages have been made not young for the past 15 seconds
mysql	innodbstatus	<code>innodb_bp_old_db_pages</code>	long	
mysql	innodbstatus	<code>innodb_bp_pages_created</code>	long_counter	The total number of pages created.
mysql	innodbstatus	<code>innodb_bp_pages_evicted_no_access</code>	double_per_sec	The average number of pages evicted without being accessed for the last 15 seconds
mysql	innodbstatus	<code>innodb_bp_pages_not_young</code>	long_counter	The number of pages made not young in the Buffer Pool LRU list
mysql	innodbstatus	<code>innodb_bp_pages_not_young_per_sec</code>	double	The per second average of the pages made young in the Buffer Pool LRU list for the last 15 seconds
mysql	innodbstatus	<code>innodb_bp_pages_read</code>	long_counter	The total number of pages read.

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodbstatus	innodb_bp_pages_read_ahead_per_sec	double	The average of read ahead operations per second for the last 15 seconds
mysql	innodbstatus	innodb_bp_pages_written	long_counter	The total number of pages written.
mysql	innodbstatus	innodb_bp_pages_young	long_counter	The number of pages made young in the Buffer Pool LRU list
mysql	innodbstatus	innodb_bp_pages_young_per_sec	double	
mysql	innodbstatus	innodb_bp_pending_pages	long	The number of pending page writes.
mysql	innodbstatus	innodb_bp_pending_writes_flushing	long_list	The number of pages to be flushed during checkpointing.
mysql	innodbstatus	innodb_bp_pending_writes_lru	long	The number of old dirty pages to be written from the bottom of the LRU list.
mysql	innodbstatus	innodb_bp_pending_writes_single_page	long	The number of pending independent page writes.
mysql	innodbstatus	innodb_bp_reads_per_sec	double	The number of buffer pool reads per second.
mysql	innodbstatus	innodb_bp_size	long	The total buffer pool size in bytes.
mysql	innodbstatus	innodb_bp_total_alloc	long	The total memory allocated for the buffer pool.
mysql	innodbstatus	innodb_bp_unzip_lru_len	long	The size in 16Kb pages of the Buffer Pool unzip_LRU list
mysql	innodbstatus	innodb_bp_written_per_sec	double	The number of buffer pool

Namespace	Namespace Type	Attribute	Type	Description
				pages written per second.
mysql	innodbstatus	<code>innodb_bp_young_hit_rate</code>	long	The rate at which pages have been made young for the past 15 seconds
mysql	innodbstatus	<code>innodb_bytes_per_read</code>	long	The number of bytes per read.
mysql	innodbstatus	<code>innodb_datetime</code>	string	The date and time the SHOW ENGINE INNODB STATUS snapshot was taken.
mysql	innodbstatus	<code>innodb_hash_node_heap</code>	long	Number of buffer pool pages reserved for the Adaptive Hash Index.
mysql	innodbstatus	<code>innodb_hash_searches_per_sec</code>	double	The number of hash searches per second.
mysql	innodbstatus	<code>innodb_hash_table_size</code>	long	The size of the hash table.
mysql	innodbstatus	<code>innodb_hash_used_cells</code>	long	Number of buffer pool pages used for the Adaptive Hash Index.
mysql	innodbstatus	<code>innodb_ibuf_free_list_len</code>	long	The length of the free list for the Insert Buffer index
mysql	innodbstatus	<code>innodb_ibuf_inserts</code>	long_counter	The number of change buffer inserts.
mysql	innodbstatus	<code>innodb_ibuf_merged_recs</code>	long_counter	The number of change buffer merged records.
mysql	innodbstatus	<code>innodb_ibuf_merges</code>	long_counter	The number of change

Namespace	Namespace Type	Attribute	Type	Description
				buffer merges.
mysql	innodbstatus	<code>innodb_ibuf_seg_size</code>	long	The Insert Buffer index header and tree size, in 16Kb page
mysql	innodbstatus	<code>innodb_ibuf_size</code>	long	The Insert Buffer index tree size, in 16Kb pages
mysql	innodbstatus	<code>innodb_io_ibuf_logs</code>	long	The number of pending log I/Os.
mysql	innodbstatus	<code>innodb_io_ibuf_reads</code>	long	The number of pending change buffer reads.
mysql	innodbstatus	<code>innodb_io_ibuf_syncs</code>	long	The number of pending synch operations.
mysql	innodbstatus	<code>innodb_io_os_file_fsyncs</code>	long_counter	The number of OS fsyncs.
mysql	innodbstatus	<code>innodb_io_os_file_reads</code>	long_counter	The number of OS file reads.
mysql	innodbstatus	<code>innodb_io_os_file_writes</code>	long_counter	The number of OS file writes.
mysql	innodbstatus	<code>innodb_io_pending_flush_bp</code>	long	The number of pending buffer pool flush operations.
mysql	innodbstatus	<code>innodb_io_pending_flush_log</code>	long	The number of pending log flush operations.
mysql	innodbstatus	<code>innodb_io_pending_reads</code>	long	The number of I/O pending reads.
mysql	innodbstatus	<code>innodb_io_pending_writes</code>	long	The number of I/O pending writes.
mysql	innodbstatus	<code>innodb_io_syncs_per_sec</code>	double	The number of fsyncs() per second.
mysql	innodbstatus	<code>innodb_log_checkpoint_file</code>	long	The log file number the last

Namespace	Namespace Type	Attribute	Type	Description
				checkpoint was performed on.
mysql	innodbstatus	<code>innodb_log_checkpoint_lsn</code>	long	The log sequence number of the last checkpoint.
mysql	innodbstatus	<code>innodb_log_checkpoint_lsn64</code>	long	The 64-bit variant of <code>innodb_log_checkpoint</code> (2.3.10+).
mysql	innodbstatus	<code>innodb_log_flushed_file</code>	long	The log file number the last checkpoint was performed on.
mysql	innodbstatus	<code>innodb_log_flushed_lsn</code>	long	The point up to which the log was last flushed.
mysql	innodbstatus	<code>innodb_log_flushed_lsn64</code>	long	The 64-bit variant of <code>innodb_log_flushed_lsn</code> (2.3.10+).
mysql	innodbstatus	<code>innodb_log_io_per_sec</code>	double	The number of log I/Os per second.
mysql	innodbstatus	<code>innodb_log_io_total</code>	long_counter	The total number of log I/Os.
mysql	innodbstatus	<code>innodb_log_pending_checkpoint_writes</code>	long	The log pending checkpoint writes.
mysql	innodbstatus	<code>innodb_log_pending_log_writes</code>	long	The log pending log writes.
mysql	innodbstatus	<code>innodb_log_sequence_file</code>	long	The log sequence file number.
mysql	innodbstatus	<code>innodb_log_sequence_lsn</code>	long	The log sequence number.
mysql	innodbstatus	<code>innodb_log_sequence_lsn64</code>	long	The 64-bit variant of <code>innodb_log_sequence</code> (2.3.10+).
mysql	innodbstatus	<code>innodb_main_thd_log_flush_writes</code>	counter	Count of log flushes

Namespace	Namespace Type	Attribute	Type	Description
				initiated within the main thread loop
mysql	innodbstatus	<code>innodb_main_thd_loops_background_counter</code>	long	Count of background loops the main thread has done
mysql	innodbstatus	<code>innodb_main_thd_loops_flush_counter</code>	long	Count of flush loops the main thread has done
mysql	innodbstatus	<code>innodb_main_thd_loops_one_second_counter</code>	long	Count of one second loops the main thread has done
mysql	innodbstatus	<code>innodb_main_thd_loops_sleep_counter</code>	long	Count of sleeps the main thread has done inside the one second loop
mysql	innodbstatus	<code>innodb_main_thd_loops_ten_second_counter</code>	long	Count of ten second loops the main thread has done
mysql	innodbstatus	<code>innodb_non_hash_searches_per_sec</code>	double	The number of nonadaptive hash index searches.
mysql	innodbstatus	<code>innodb_per_sec_avg</code>	double	The number of seconds the averages for SHOW INNODB STATUS were calculated from.
mysql	innodbstatus	<code>innodb_reads_per_sec</code>	double	The number of reads per second.
mysql	innodbstatus	<code>innodb_row_queries_inside</code>	long	The number of queries executing inside InnoDB.
mysql	innodbstatus	<code>innodb_row_queries_queue</code>	long	The number of queries in the queue,

Namespace	Namespace Type	Attribute	Type	Description
				waiting to enter InnoDB.
mysql	innodbstatus	<code>innodb_row_state</code>	<code>string</code>	The current state of the main InnoDB thread.
mysql	innodbstatus	<code>innodb_rows_deleted</code>	<code>long_counter</code>	The number of rows deleted from InnoDB tables.
mysql	innodbstatus	<code>innodb_rows_deleted_per_sec</code>	<code>double</code>	The number of rows deleted per second from InnoDB tables.
mysql	innodbstatus	<code>innodb_rows_inserted</code>	<code>long_counter</code>	The number of rows inserted into InnoDB tables.
mysql	innodbstatus	<code>innodb_rows_inserted_per_sec</code>	<code>double</code>	The number of rows inserted per second into InnoDB tables.
mysql	innodbstatus	<code>innodb_rows_read</code>	<code>long_counter</code>	The number of rows read from InnoDB tables.
mysql	innodbstatus	<code>innodb_rows_read_per_sec</code>	<code>double</code>	The number of rows read per second from InnoDB tables.
mysql	innodbstatus	<code>innodb_rows_updated</code>	<code>long_counter</code>	The number of rows updated in InnoDB tables.
mysql	innodbstatus	<code>innodb_rows_updated_per_sec</code>	<code>double</code>	The number of rows updated per second in InnoDB tables.
mysql	innodbstatus	<code>innodb_sem_mutex_os_waits</code>	<code>long_counter</code>	The number of semaphore/mutex waits

Namespace	Namespace Type	Attribute	Type	Description
				yielded to the OS.
mysql	innodbstatus	<code>innodb_sem_mutex_rounds</code>	<code>long_counter</code>	The number of semaphore/mutex round for the internal sync array.
mysql	innodbstatus	<code>innodb_sem_mutex_spin_waits</code>	<code>long_counter</code>	The number of semaphore/mutexspin waits for the internal sync array.
mysql	innodbstatus	<code>innodb_sem_os_reservation_count</code>	<code>long_counter</code>	The number of times a mutex wait was added to the internal sync array.
mysql	innodbstatus	<code>innodb_sem_os_signal_count</code>	<code>long_counter</code>	Performance.InnoDB
mysql	innodbstatus	<code>innodb_sem_rw_excl_os_waits</code>	<code>long_counter</code>	The number of exclusive (write) semaphore waits yielded to the OS.
mysql	innodbstatus	<code>innodb_sem_rw_excl_rounds</code>	<code>long_counter</code>	Performance.InnoDB
mysql	innodbstatus	<code>innodb_sem_rw_excl_spins</code>	<code>long_counter</code>	Performance.InnoDB
mysql	innodbstatus	<code>innodb_sem_shared_os_waits</code>	<code>long_counter</code>	The number of shared (read) semaphore waits yielded to the OS.
mysql	innodbstatus	<code>innodb_sem_shared_rounds</code>	<code>long_counter</code>	Performance.InnoDB
mysql	innodbstatus	<code>innodb_sem_shared_spins</code>	<code>long_counter</code>	The number of shared (read) semaphore spin waits within the sync array.
mysql	innodbstatus	<code>innodb_sem_spins_per_wait_mutex</code>	<code>double</code>	Ratio of <code>innodb_sem_mutex_rounds</code> to <code>innodb_sem_mutex_spin_waits</code> .
mysql	innodbstatus	<code>innodb_sem_spins_per_wait_rwlock</code>	<code>double</code>	Ratio of RW-Exclusive spin rounds to <code>innodb_sem_rw_excl_spins</code> .

Namespace	Namespace Type	Attribute	Type	Description
mysql	innodbstatus	innodb_sem_spins_per_wait_double	double	Ratio of RW-Shared spin rounds to innodb_sem_shared_s
mysql	innodbstatus	innodb_trx_history_list_length	long	The transaction history list length.
mysql	innodbstatus	innodb_trx_id_counter	long_counter	Decoded decimal representation of the hex transaction ID counter in the latest InnoDB
mysql	innodbstatus	innodb_trx_id_counter1	long	The transaction counter roll over variable.
mysql	innodbstatus	innodb_trx_id_counter2	long	The main transaction count.
mysql	innodbstatus	innodb_trx_id_counter_str	string	The hex string based transaction ID counter in the latest InnoDB
mysql	innodbstatus	innodb_trx_purge_done_trx2	long	The transaction count that is already purged.
mysql	innodbstatus	innodb_trx_total_lock_structs	long	
mysql	innodbstatus	innodb_writes_per_sec	double	The number of InnoDB writes per second.
mysql	locked_processes	num_locked	long	
mysql	long_locked_processes	ses_long_running_time	double	
mysql	long_locked_processes	ses_long_running	long	
mysql	long_running_processes	ses_long_running_time	double	
mysql	long_running_processes	ses_long_running	long	
mysql	masterlogs	File_size	long	The size of a binary log file.
mysql	masterlogs	filecount	long	The number of binary log files.
mysql	masterlogs	filesizesum	long	The total size of all the

Namespace	Namespace Type	Attribute	Type	Description
				binary log files.
mysql	masterlogs	Log_name	string	The name of the binary log file.
mysql	MasterStatus	Binlog_Do_DB	string	
mysql	MasterStatus	Binlog_Ignore_DB	string	
mysql	MasterStatus	File	string	
mysql	MasterStatus	Position	long	
mysql	meb_backup_history	backup_destination	string	
mysql	meb_backup_history	collected_ts	integer	
mysql	meb_backup_history	collected_ts_counter	integer	
mysql	meb_backup_history	end_time_raw	string	
mysql	meb_backup_history	end_time_ts	integer	
mysql	meb_backup_history	exit_state	string	
mysql	meb_backup_history	last_error	string	
mysql	meb_backup_history	last_error_code	integer	
mysql	meb_backup_history	lock_time	float	
mysql	meb_backup_history	mysql_data_dir	string	
mysql	meb_backup_history	progress_log	string	
mysql	meb_backup_history	start_time_now	string	
mysql	meb_backup_history	start_time_ts	integer	
mysql	meb_backup_history	total_time	integer	
mysql	meb_backup_success_history	total_all_backup	string	
mysql	meb_incremental	total_increments	integer	
mysql	meb_timing	lock_time	float	
mysql	meb_timing	total_time	integer	
mysql	no_password	user	string	
mysql	no_root_password	no_password	long	
mysql	num_waiting_txns	num_waiting_txns	long	
mysql	prepared_count	prepared_count	long	
mysql	privileges_on_all_dbs	user_spec	string	
mysql	processlist	Command	string	The type of command the thread is executing.
mysql	processlist	db	string	The default database, if one is selected.
mysql	processlist	Host	string	The host name of the client issuing the statement.

Namespace	Namespace Type	Attribute	Type	Description
mysql	processlist	Id	long	The connection identifier.
mysql	processlist	Info	string	The statement that the thread is executing.
mysql	processlist	State	string	An action, event, or state that indicates what the thread is doing.
mysql	processlist	Time	long	The time in seconds that the thread has been in its current state.
mysql	processlist	User	string	The MySQL user who issued the statement.
mysql	repl_slave_privileges	repl_slaves	long	
mysql	rolledback_count	rolledback_count	long	
mysql	root_account_exists	root_account	long	
mysql	root_remote_login	remote_login	long	
mysql	Server	blackout	string	
mysql	Server	displayname	string	The display name of the server in the Dashboard.
mysql	Server	registration-complete	string	
mysql	Server	repl.group	long	
mysql	Server	repl.groupName	string	
mysql	Server	server.connected	long	Whether the server is connected.
mysql	Server	server.last_errno	string	
mysql	Server	server.last_error	string	The last MySQL server error message.
mysql	Server	server.reachable	long	Whether the server is reachable.
mysql	Server	server.version_numeric	long	The MySQL server version number.
mysql	Server	Time	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	Server	transport	string	
mysql	Server	visible.displayname	string	
mysql	server_admin_privileges	server_spec	string	
mysql	SlaveStatus	Binlog_Do_DB	string	
mysql	SlaveStatus	Binlog_Ignore_DB	string	
mysql	SlaveStatus	Connect_Retry	long	
mysql	SlaveStatus	Exec_Master_Log_Pos	long	
mysql	SlaveStatus	File	string	
mysql	SlaveStatus	Last_Errno	long	
mysql	SlaveStatus	Last_Error	string	
mysql	SlaveStatus	Last_IO_Errno	long	
mysql	SlaveStatus	Last_IO_Error	string	
mysql	SlaveStatus	Last_SQL_Errno	long	
mysql	SlaveStatus	Last_SQL_Error	string	
mysql	SlaveStatus	Master_Host	string	
mysql	SlaveStatus	Master_ip	string	
mysql	SlaveStatus	Master_Log_File	string	
mysql	SlaveStatus	Master_Port	long	
mysql	SlaveStatus	Master_SSL_Allowed	string	
mysql	SlaveStatus	Master_SSL_CA_File	string	
mysql	SlaveStatus	Master_SSL_CA_Path	string	
mysql	SlaveStatus	Master_SSL_Cert	string	
mysql	SlaveStatus	Master_SSL_Cipher	string	
mysql	SlaveStatus	Master_SSL_Key	string	
mysql	SlaveStatus	Master_SSL_Verify_Server_Cert	string	
mysql	SlaveStatus	Master_User	string	
mysql	SlaveStatus	Master_uuid	string	
mysql	SlaveStatus	Position	long	
mysql	SlaveStatus	Read_Master_Log_Pos	long	
mysql	SlaveStatus	Relay_Log_File	string	
mysql	SlaveStatus	Relay_Log_Pos	long	
mysql	SlaveStatus	Relay_Log_Space	long	
mysql	SlaveStatus	Relay_Master_Log_File	string	
mysql	SlaveStatus	Replicate_Do_DB	string	
mysql	SlaveStatus	Replicate_Do_Table	string	
mysql	SlaveStatus	Replicate_Ignore_DB	string	
mysql	SlaveStatus	Replicate_Ignore_Table	string	
mysql	SlaveStatus	Replicate_Wild_Do_Table	string	
mysql	SlaveStatus	Replicate_Wild_Ignore_Table	string	
mysql	SlaveStatus	Seconds_Behind_Master	long	
mysql	SlaveStatus	Skip_Counter	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	SlaveStatus	Slave_IO_Running	string	
mysql	SlaveStatus	Slave_IO_State	string	
mysql	SlaveStatus	Slave_SQL_Running	string	
mysql	SlaveStatus	Until_Condition	string	
mysql	SlaveStatus	Until_Log_File	string	
mysql	SlaveStatus	Until_Log_Pos	long	
mysql	sp_with_select_star	routine	string	
mysql	Statement	bytes	long	
mysql	Statement	comment	string	
mysql	Statement	connection_id	long	
mysql	Statement	database	string	
mysql	Statement	errors	long	
mysql	Statement	exec_time	long	
mysql	Statement	explain_plan	string	
mysql	Statement	host_from	string	
mysql	Statement	host_to	string	
mysql	Statement	no_good_index_used	long	
mysql	Statement	no_index_used	long	
mysql	Statement	query_type	string	
mysql	Statement	rows	long	
mysql	Statement	source_location	string	
mysql	Statement	text	string	
mysql	Statement	user	string	
mysql	Statement	warnings	long	
mysql	StatementAnalysisSupport	explain_min_exec_time_max	long	
mysql	StatementAnalysisSupport	profile_examples	string	
mysql	StatementAnalysisSupport	profile_explain	string	
mysql	StatementAnalysisSupport	profile	string	
mysql	StatementAnalysisSupport	frequency	string	
mysql	StatementSummary	avg_exec_time	string	
mysql	StatementSummary	bytes	long	
mysql	StatementSummary	bytes_rel	long	
mysql	StatementSummary	count	long	
mysql	StatementSummary	count_rel	long	
mysql	StatementSummary	database	string	
mysql	StatementSummary	errors	long	
mysql	StatementSummary	exec_time	long	
mysql	StatementSummary	exec_time_rel	long	
mysql	StatementSummary	max_bytes	long	
mysql	StatementSummary	max_exec_time	long	
mysql	StatementSummary	max_rows	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	StatementSummary	min_bytes	long	
mysql	StatementSummary	min_exec_time	long	
mysql	StatementSummary	min_rows	long	
mysql	StatementSummary	no_good_index_used	long	
mysql	StatementSummary	no_index_used	long	
mysql	StatementSummary	query_type	string	
mysql	StatementSummary	rows	long	
mysql	StatementSummary	rows_rel	long	
mysql	StatementSummary	text	string	
mysql	StatementSummary	text_hash	string	
mysql	StatementSummary	warnings	long	
mysql	status	Aborted_clients	long_counter	Networking.Overview
mysql	status	Aborted_connects	long_counter	The number of failed attempts to connect to the MySQL server.
mysql	status	Binlog_cache_disk_use	long_counter	General.Logging
mysql	status	Binlog_cache_use	long_counter	The number of transactions that used the temporary binary log cache.
mysql	status	Bytes_received	long_counter	The number of bytes received from all clients.
mysql	status	Bytes_sent	long_counter	The number of bytes sent to all clients.
mysql	status	Com_admin_commands	long_counter	Count of admin commands.
mysql	status	Com_alter_db	long_counter	Count of ALTER DATABASE statements.
mysql	status	Com_alter_db_upgrade	long_counter	
mysql	status	Com_alter_event	long_counter	
mysql	status	Com_alter_function	long_counter	
mysql	status	Com_alter_procedure	long_counter	
mysql	status	Com_alter_server	long_counter	
mysql	status	Com_alter_table	long_counter	Count of ALTER

Namespace	Namespace Type	Attribute	Type	Description
				TABLE statements.
mysql	status	Com_alter_tablespace	long_counter	
mysql	status	Com_analyze	long_counter	Count of ANALYZE statements.
mysql	status	Com_assign_to_keycache	long_counter	
mysql	status	Com_backup_table	long_counter	Count of BACKUP TABLE statements.
mysql	status	Com_begin	long_counter	Count of BEGIN statements.
mysql	status	Com_binlog	long_counter	
mysql	status	Com_call_procedure	long_counter	Number of calls to stored procedures.
mysql	status	Com_change_db	long_counter	Count of CHANGE DATABASE statements.
mysql	status	Com_change_master	long_counter	Count of CHANGE MASTER statements.
mysql	status	Com_check	long_counter	Count of CHECK statements.
mysql	status	Com_checksum	long_counter	Count of CHECKSUM statements.
mysql	status	Com_commit	long_counter	Count of COMMIT statements.
mysql	status	Com_create_db	long_counter	Count of CREATE DATABASE statements.
mysql	status	Com_create_event	long_counter	
mysql	status	Com_create_function	long_counter	Count of CREATE FUNCTION statements.
mysql	status	Com_create_index	long_counter	Count of CREATE INDEX statements.
mysql	status	Com_create_procedure	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_create_server	long_counter	
mysql	status	Com_create_table	long_counter	Count of CREATE TABLE statements.
mysql	status	Com_create_trigger	long_counter	
mysql	status	Com_create_udf	long_counter	
mysql	status	Com_create_user	long_counter	Count of CREATE USER statements.
mysql	status	Com_create_view	long_counter	
mysql	status	Com_dealloc_sql	long_counter	Count of DEALLOCATE PREPARE statements.
mysql	status	Com_delete	long_counter	Count of DELETE statements.
mysql	status	Com_delete_multi	long_counter	Count of multi-table DELETE statements.
mysql	status	Com_do	long_counter	Count of DO statements.
mysql	status	Com_drop_db	long_counter	Count of DROP DATABASE statements.
mysql	status	Com_drop_event	long_counter	
mysql	status	Com_drop_function	long_counter	Count of DROP FUNCTION statements.
mysql	status	Com_drop_index	long_counter	Count of DROP INDEX statements.
mysql	status	Com_drop_procedure	long_counter	
mysql	status	Com_drop_server	long_counter	
mysql	status	Com_drop_table	long_counter	Count of DROP TABLE statements.
mysql	status	Com_drop_trigger	long_counter	
mysql	status	Com_drop_user	long_counter	Count of DROP USER statements.
mysql	status	Com_drop_view	long_counter	
mysql	status	Com_empty_query	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_execute_sql	long_counter	Count of EXECUTE statements.
mysql	status	Com_flush	long_counter	Count of FLUSH statements.
mysql	status	Com_grant	long_counter	Count of GRANT statements.
mysql	status	Com_ha_close	long_counter	Count of HANDLER CLOSE statements.
mysql	status	Com_ha_open	long_counter	Count of HANDLER OPEN statements.
mysql	status	Com_ha_read	long_counter	Count of HANDLER READ statements.
mysql	status	Com_help	long_counter	Count of HELP statements.
mysql	status	Com_insert	long_counter	Count of INSERT statements.
mysql	status	Com_insert_select	long_counter	Count of INSERT SELECT statements.
mysql	status	Com_install_plugin	long_counter	
mysql	status	Com_kill	long_counter	Count of KILL statements.
mysql	status	Com_load	long_counter	Count of LOAD statements.
mysql	status	Com_load_master_data	long_counter	Count of LOAD MASTER DATA statements.
mysql	status	Com_load_master_table	long_counter	Count of LOAD MASTER TABLE statements.
mysql	status	Com_lock_tables	long_counter	Count of LOCK TABLES statements.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_optimize	long_counter	Count of OPTIMIZE statements.
mysql	status	Com_preload_keys	long_counter	Count of PRELOAD KEYS statements.
mysql	status	Com_prepare_sql	long_counter	Count of PREPARE statements.
mysql	status	Com_purge	long_counter	Count of PURGE statements.
mysql	status	Com_purge_before_date	long_counter	Count of PURGE BEFORE DATE statements.
mysql	status	Com_release_savepoint	long_counter	
mysql	status	Com_rename_table	long_counter	Count of RENAME TABLE statements.
mysql	status	Com_rename_user	long_counter	
mysql	status	Com_repair	long_counter	Count of REPAIR statements.
mysql	status	Com_replace	long_counter	Count of REPLACE statements.
mysql	status	Com_replace_select	long_counter	Count of REPLACE SELECT statements.
mysql	status	Com_reset	long_counter	Count of RESET statements.
mysql	status	Com_restore_table	long_counter	Count of RESTORE TABLE statements.
mysql	status	Com_revoke	long_counter	Count of REVOKE statements.
mysql	status	Com_revoke_all	long_counter	Count of REVOKE ALL statements.
mysql	status	Com_rollback	long_counter	Count of ROLLBACK statements.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_rollback_to_savepoint	long_counter	
mysql	status	Com_savepoint	long_counter	Count of SAVEPOINT statements.
mysql	status	Com_select	long_counter	Count of SELECT statements.
mysql	status	Com_set_option	long_counter	Count of SET OPTION statements.
mysql	status	Com_show_authors	long_counter	
mysql	status	Com_show_binlog_events	long_counter	Count of SHOW BINLOG EVENTS statements.
mysql	status	Com_show_binlogs	long_counter	Count of SHOW BINLOGS statements.
mysql	status	Com_show_charsets	long_counter	Count of SHOW CHARSETS statements.
mysql	status	Com_show_collations	long_counter	Count of SHOW COLLATIONS statements.
mysql	status	Com_show_column_types	long_counter	Count of SHOW COLUMN TYPES statements.
mysql	status	Com_show_contributors	long_counter	
mysql	status	Com_show_create_db	long_counter	Count of SHOW CREATE DATABASE statements.
mysql	status	Com_show_create_event	long_counter	
mysql	status	Com_show_create_func	long_counter	
mysql	status	Com_show_create_proc	long_counter	
mysql	status	Com_show_create_table	long_counter	Count of SHOW CREATE TABLE statements.
mysql	status	Com_show_create_trigger	long_counter	
mysql	status	Com_show_databases	long_counter	Count of SHOW

Namespace	Namespace Type	Attribute	Type	Description
				DATABASES statements.
mysql	status	Com_show_engine_logs	long_counter	
mysql	status	Com_show_engine_mutex	long_counter	
mysql	status	Com_show_engine_status	long_counter	
mysql	status	Com_show_errors	long_counter	Count of SHOW ERRORS statements.
mysql	status	Com_show_events	long_counter	
mysql	status	Com_show_fields	long_counter	Count of SHOW FIELDS statements.
mysql	status	Com_show_function_status	long_counter	
mysql	status	Com_show_grants	long_counter	Count of SHOW GRANTS statements.
mysql	status	Com_show_innodb_status	long_counter	Count of SHOW INNODB STATUS statements.
mysql	status	Com_show_keys	long_counter	Count of SHOW KEYS statements.
mysql	status	Com_show_logs	long_counter	Count of SHOW LOGS statements.
mysql	status	Com_show_master_status	long_counter	Count of SHOW MASTER STATUS statements.
mysql	status	Com_show_ndb_status	long_counter	Count of SHOW NDB STATUS statements.
mysql	status	Com_show_new_master	long_counter	Count of SHOW NEW MASTER statements.
mysql	status	Com_show_open_tables	long_counter	Count of SHOW OPEN TABLES statements.
mysql	status	Com_show_plugins	long_counter	
mysql	status	Com_show_privileges	long_counter	Count of SHOW

Namespace	Namespace Type	Attribute	Type	Description
				PRIVILEGES statements.
mysql	status	Com_show_procedure_status	long_counter	
mysql	status	Com_show_processlist	long_counter	Count of SHOW PROCESSLIST statements.
mysql	status	Com_show_profile	long_counter	
mysql	status	Com_show_profiles	long_counter	
mysql	status	Com_show_slave_hosts	long_counter	Count of SHOW SLAVE HOSTS statements.
mysql	status	Com_show_slave_status	long_counter	Count of SHOW SLAVE STATUS statements.
mysql	status	Com_show_status	long_counter	Count of SHOW STATUS statements.
mysql	status	Com_show_storage_engines	long_counter	Count of SHOW STORAGE ENGINES statements.
mysql	status	Com_show_table_status	long_counter	
mysql	status	Com_show_tables	long_counter	Count of SHOW TABLES statements.
mysql	status	Com_show_triggers	long_counter	Count of SHOW TRIGGERS statements.
mysql	status	Com_show_variables	long_counter	Count of SHOW VARIABLES statements.
mysql	status	Com_show_warnings	long_counter	Count of SHOW WARNINGS statements.
mysql	status	Com_slave_start	long_counter	Count of START SLAVE statements.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_slave_stop	long_counter	Count of STOP SLAVE statements.
mysql	status	Com_stmt_close	long_counter	Count of STATEMENT CLOSE statements.
mysql	status	Com_stmt_execute	long_counter	Count of STATEMENT EXECUTE statements.
mysql	status	Com_stmt_fetch	long_counter	Count of STATEMENT FETCH statements.
mysql	status	Com_stmt_prepare	long_counter	Count of STATEMENT PREPARE statements.
mysql	status	Com_stmt_reprepare	long_counter	
mysql	status	Com_stmt_reset	long_counter	Count of STATEMENT RESET statements.
mysql	status	Com_stmt_send_long_data	long_counter	Count of STATEMENT SEND LONG DATA statements.
mysql	status	Com_truncate	long_counter	Count of TRUNCATE statements.
mysql	status	Com_uninstall_plugin	long_counter	
mysql	status	Com_unlock_tables	long_counter	Count of UNLOCK TABLES statements.
mysql	status	Com_update	long_counter	Count of UPDATE statements.
mysql	status	Com_update_multi	long_counter	Count of multi-table UPDATE statements.
mysql	status	Com_xa_commit	long_counter	Count of XA COMMIT statements.
mysql	status	Com_xa_end	long_counter	Count of XA END statements.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Com_xa_prepare	long_counter	Count of XA PREPARE statements.
mysql	status	Com_xa_recover	long_counter	Count of XA RECOVER statements.
mysql	status	Com_xa_rollback	long_counter	Count of XA ROLLBACK statements.
mysql	status	Com_xa_start	long_counter	Count of XA START statements.
mysql	status	Compression	string	General.Features
mysql	status	Connections	long_counter	The number of connection attempts.
mysql	status	Created_tmp_disk_tables	long_counter	General.Temporary
mysql	status	Created_tmp_files	long_counter	How many temporary files mysqld has created.
mysql	status	Created_tmp_tables	long_counter	How many temporary tables mysqld has created.
mysql	status	Delayed_errors	long_counter	Performance.Delayed
mysql	status	Delayed_insert_threads	long	The number of INSERT DELAYED thread handlers in use. For non-transactional tables only.
mysql	status	Delayed_writes	long_counter	The number of INSERT DELAYED rows written. For non-transactional tables only.
mysql	status	displayname	string	The display name of the server in the Dashboard.
mysql	status	Flush_commands	long_counter	The number of FLUSH statements executed.
mysql	status	groupname	string	The name of the group

Namespace	Namespace Type	Attribute	Type	Description
				to which the server belongs.
mysql	status	<code>Handler_commit</code>	<code>long_counter</code>	The number of internal COMMIT statements.
mysql	status	<code>Handler_delete</code>	<code>long_counter</code>	The number of times that rows have been deleted from tables.
mysql	status	<code>Handler_discover</code>	<code>long_counter</code>	The number of times that tables have been discovered.
mysql	status	<code>Handler_prepare</code>	<code>long_counter</code>	A counter for the prepare phase of two-phase commit operations.
mysql	status	<code>Handler_read_first</code>	<code>long_counter</code>	The number of times the first entry was read from an index.
mysql	status	<code>Handler_read_key</code>	<code>long_counter</code>	The number of requests to read a row based on a key.
mysql	status	<code>Handler_read_next</code>	<code>long_counter</code>	The number of requests to read the next row in key order.
mysql	status	<code>Handler_read_prev</code>	<code>long_counter</code>	The number of requests to read the previous row in key order.
mysql	status	<code>Handler_read_rnd</code>	<code>long_counter</code>	The number of requests to read a row based on a fixed position.
mysql	status	<code>Handler_read_rnd_next</code>	<code>long_counter</code>	The number of requests to read the next row in the data file.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Handler_rollback	long_counter	Miscellaneous. Handler
mysql	status	Handler_savepoint	long_counter	The number of requests for a storage engine to place a savepoint.
mysql	status	Handler_savepoint_rollback	long_counter	The number of requests for a storage engine to roll back to a savepoint.
mysql	status	Handler_update	long_counter	The number of requests to update a row in a table.
mysql	status	Handler_write	long_counter	The number of requests to insert a row in a table.
mysql	status	Innodb_buffer_pool_pages_data	long	The number of pages containing data (dirty or clean)
mysql	status	Innodb_buffer_pool_pages_dirty	long	The number of pages currently dirty in the InnoDB buffer pool.
mysql	status	Innodb_buffer_pool_pages_flushed	long_counter	The number of page flush requests in the InnoDB buffer pool.
mysql	status	Innodb_buffer_pool_pages_free	long	The number of InnoDB buffer pool pages free.
mysql	status	Innodb_buffer_pool_pages_latched	long	The number of latched pages in InnoDB buffer pool.
mysql	status	Innodb_buffer_pool_pages_mixed	long	Performance. InnoDB
mysql	status	Innodb_buffer_pool_pages_total	long	The total size of the InnoDB buffer pool, in pages.
mysql	status	Innodb_buffer_pool_read_ahead	long_counter	The number of random

Namespace	Namespace Type	Attribute	Type	Description
				read-aheads initiated by InnoDB.
mysql	status	InnoDB_buffer_pool_read_aheads	long_counter	The number of sequential read-aheads initiated by InnoDB.
mysql	status	InnoDB_buffer_pool_read_requests	long_counter	The number of logical read requests InnoDB has done.
mysql	status	InnoDB_buffer_pool_reads	long_counter	Performance.InnoDB
mysql	status	InnoDB_buffer_pool_wait_free	long_counter	Number of waits for pages to be flushed.
mysql	status	InnoDB_buffer_pool_write_requests	long_counter	The number of writes done to the InnoDB buffer pool.
mysql	status	InnoDB_data_fsyncs	long_counter	The number of fsync() operations so far.
mysql	status	InnoDB_data_pending_fsyncs	long	The current number of pending fsync() operations.
mysql	status	InnoDB_data_pending_reads	long	The current number of pending reads.
mysql	status	InnoDB_data_pending_writes	long	The number of pending writes.
mysql	status	InnoDB_data_read	long_counter	The amount of data read so far, in bytes.
mysql	status	InnoDB_data_reads	long_counter	The total number of data reads.
mysql	status	InnoDB_data_writes	long_counter	The total number of data writes.
mysql	status	InnoDB_data_written	long_counter	The amount of data written in bytes.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Innodb_dblwr_pages_written	long_counter	The number of doublewrite pages that have been written.
mysql	status	Innodb_dblwr_writes	long_counter	The number of doublewrite operations that have been performed.
mysql	status	Innodb_log_waits	long_counter	Performance.InnoDB
mysql	status	Innodb_log_write_requests	long_counter	The number of log write requests.
mysql	status	Innodb_log_writes	long_counter	The number of physical writes to the log.
mysql	status	Innodb_os_log_fsyncs	long_counter	The number of fsync() writes done to the log file.
mysql	status	Innodb_os_log_pending_fsyncs	long	The number of pending log file fsync() operations.
mysql	status	Innodb_os_log_pending_writes	long	The number of pending log file writes.
mysql	status	Innodb_os_log_written	long_counter	The number of bytes written to the log file.
mysql	status	Innodb_page_size	long	The compiled-in InnoDB page size.
mysql	status	Innodb_pages_created	long_counter	The number of pages created.
mysql	status	Innodb_pages_read	long_counter	The number of pages read.
mysql	status	Innodb_pages_written	long_counter	The number of pages written.
mysql	status	Innodb_row_lock_current_waits	long	The number of row locks currently being waited for.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	<code>Innodb_row_lock_time</code>	<code>long_counter</code>	The total time spent in acquiring row locks, in milliseconds.
mysql	status	<code>Innodb_row_lock_time_avg</code>	<code>long</code>	The average time to acquire a row lock, in milliseconds.
mysql	status	<code>Innodb_row_lock_time_max</code>	<code>long</code>	The maximum time to acquire a row lock, in milliseconds.
mysql	status	<code>Innodb_row_lock_waits</code>	<code>long_counter</code>	The number of times a row lock had to be waited for.
mysql	status	<code>Innodb_rows_deleted</code>	<code>long_counter</code>	The number of rows deleted from InnoDB tables.
mysql	status	<code>Innodb_rows_inserted</code>	<code>long_counter</code>	The number of rows inserted into InnoDB tables.
mysql	status	<code>Innodb_rows_read</code>	<code>long_counter</code>	The number of rows read from InnoDB tables.
mysql	status	<code>Innodb_rows_updated</code>	<code>long_counter</code>	The number of rows updated in InnoDB tables.
mysql	status	<code>Key_blocks_not_flushed</code>	<code>long</code>	Performance.Keys
mysql	status	<code>Key_blocks_unused</code>	<code>long</code>	The number of unused blocks in the key cache.
mysql	status	<code>Key_blocks_used</code>	<code>long</code>	The number of used blocks in the key cache.
mysql	status	<code>Key_read_requests</code>	<code>long_counter</code>	The number of requests to read a key block from the cache.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Key_reads	long_counter	The number of physical reads of a key block from disk.
mysql	status	Key_write_requests	long_counter	The number of requests to write a key block to the cache.
mysql	status	Key_writes	long_counter	The number of physical writes of a key block to disk.
mysql	status	Last_query_cost	long	Performance.Cost
mysql	status	Max_used_connections	long	Networking.Overview
mysql	status	Ndb_number_of_data_nodes	long	
mysql	status	Ndb_number_of_ready_data_nodes	long	
mysql	status	Not_flushed_delayed_rows	long	The number of rows waiting to be written in INSERT DELAY queues. For non-transactional tables only.
mysql	status	Open_files	long	The number of files that are open.
mysql	status	Open_streams	long	The number of streams that are open (used mainly for logging)
mysql	status	Open_table_definitions	long	
mysql	status	Open_tables	long	The number of tables that are open.
mysql	status	Opened_files	long_counter	
mysql	status	Opened_table_definitions	long_counter	
mysql	status	Opened_tables	long_counter	The number of tables that have been opened.
mysql	status	Prepared_stmt_count	long	The current number of prepared statements.

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	<code>Qcache_free_blocks</code>	long	The number of free memory blocks in the query cache.
mysql	status	<code>Qcache_free_memory</code>	long	The amount of free memory for the query cache.
mysql	status	<code>Qcache_hits</code>	long_counter	The number of query cache hits.
mysql	status	<code>Qcache_inserts</code>	long_counter	The number of query cache inserts.
mysql	status	<code>Qcache_lowmem_prunes</code>	long_counter	Performance.Query Cache
mysql	status	<code>Qcache_not_cached</code>	long_counter	Performance.Query Cache
mysql	status	<code>Qcache_queries_in_cache</code>	long	The number of queries registered in the query cache.
mysql	status	<code>Qcache_total_blocks</code>	long	The total number of blocks in the query cache.
mysql	status	<code>Queries</code>	long_counter	
mysql	status	<code>Questions</code>	long_counter	The number of statements that clients have sent to the server.
mysql	status	<code>Rpl_status</code>	long	The status of fail-safe replication (not yet implemented)
mysql	status	<code>Select_full_join</code>	long_counter	Performance.Selects
mysql	status	<code>Select_full_range_join</code>	long_counter	The number of joins that used a range search on a reference table.
mysql	status	<code>Select_range</code>	long_counter	The number of joins that used ranges

Namespace	Namespace Type	Attribute	Type	Description
				on the first table.
mysql	status	Select_range_check	long_counter	The number of joins without keys that check for key usage after each row.
mysql	status	Select_scan	long_counter	The number of joins that did a full scan of the first table.
mysql	status	Slave_open_temp_tables	long	Networking.Replication
mysql	status	Slave_retried_transactions	long_counter	Networking.Replication
mysql	status	Slave_running	string	This is ON if this server is a slave that is connected to a master.
mysql	status	Slow_launch_threads	long_counter	Performance.Slow Items
mysql	status	Slow_queries	long_counter	The number of queries that have taken more than long_query_time seconds.
mysql	status	Sort_merge_passes	long_counter	The number of merge passes that the sort algorithm has had to do.
mysql	status	Sort_range	long_counter	The number of sorts that were done using ranges.
mysql	status	Sort_rows	long_counter	The number of sorted rows.
mysql	status	Sort_scan	long_counter	The number of sorts that were done by scanning the table.
mysql	status	Ssl_accept_renegotiates	long_counter	The number of renegotiates needed to

Namespace	Namespace Type	Attribute	Type	Description
				establish the connection.
mysql	status	<code>Ssl_accepts</code>	<code>long_counter</code>	The number of attempted SSL connections.
mysql	status	<code>Ssl_callback_cache_hits</code>	<code>long_counter</code>	The number of callback cache hits.
mysql	status	<code>Ssl_cipher</code>	<code>string</code>	SSL cipher to use (implies --ssl)
mysql	status	<code>Ssl_cipher_list</code>	<code>string</code>	The list of SSL cipher strings.
mysql	status	<code>Ssl_client_connects</code>	<code>long_counter</code>	The number of attempted connections to an SSL-enabled master.
mysql	status	<code>Ssl_connect_renegotiates</code>	<code>long_counter</code>	Number of renegotiates needed to connect to an SSL-enabled master.
mysql	status	<code>Ssl_ctx_verify_depth</code>	<code>long</code>	The SSL context verification depth.
mysql	status	<code>Ssl_ctx_verify_mode</code>	<code>long</code>	The SSL certificate verification mode used by the server.
mysql	status	<code>Ssl_default_timeout</code>	<code>long</code>	The SSL default timeout.
mysql	status	<code>Ssl_finished_accepts</code>	<code>long_counter</code>	The number of successful SSL connections to the server.
mysql	status	<code>Ssl_finished_connects</code>	<code>long_counter</code>	The number of successful SSL slave connections to the server.
mysql	status	<code>Ssl_session_cache_hits</code>	<code>long_counter</code>	The number of SSL

Namespace	Namespace Type	Attribute	Type	Description
				session cache hits.
mysql	status	<code>Ssl_session_cache_misses</code>	<code>long_counter</code>	The number of SSL session cache misses.
mysql	status	<code>Ssl_session_cache_mode</code>	<code>string</code>	The SSL session cache mode.
mysql	status	<code>Ssl_session_cache_overflows</code>	<code>long_counter</code>	The number of SSL cache overflows.
mysql	status	<code>Ssl_session_cache_size</code>	<code>long</code>	The size of the SSL session cache.
mysql	status	<code>Ssl_session_cache_timeouts</code>	<code>long_counter</code>	The number of session cache timeouts.
mysql	status	<code>Ssl_sessions_reused</code>	<code>long</code>	The number of SSL sessions from the cache that were reused.
mysql	status	<code>Ssl_used_session_cache_entries</code>	<code>long</code>	The number of SSL session cache entries used.
mysql	status	<code>Ssl_verify_depth</code>	<code>long</code>	Depth of verification for replication SSL connections.
mysql	status	<code>Ssl_verify_mode</code>	<code>long</code>	Mode of verification for replication SSL connections.
mysql	status	<code>Ssl_version</code>	<code>string</code>	The SSL version number.
mysql	status	<code>Table_locks_immediate</code>	<code>long_counter</code>	The number of times that a table lock was acquired immediately.
mysql	status	<code>Table_locks_waited</code>	<code>long_counter</code>	Performance.Locks
mysql	status	<code>Tc_log_max_pages_used</code>	<code>long</code>	General.Logging
mysql	status	<code>Tc_log_page_size</code>	<code>long</code>	General.Logging

Namespace	Namespace Type	Attribute	Type	Description
mysql	status	Tc_log_page_waits	long	General.Logging
mysql	status	Threads_cached	long	The number of threads in the thread cache.
mysql	status	Threads_connected	long	The number of currently open connections.
mysql	status	Threads_created	long_counter	The number of threads created to handle connections.
mysql	status	Threads_running	long	The number of threads that are not sleeping.
mysql	status	Uptime	long	The number of seconds the server has been up.
mysql	status	Uptime_since_flush_status	long_counter	
mysql	strong_privileges	user	string	
mysql	table	num_rows	long	MySQL Network.Table Data
mysql	table	table_auto_increment	long	The next AUTO_INCREMENT value.
mysql	table	table_avg_row_length	long	The average row length.
mysql	table	table_collation	string	The table's character set and collation.
mysql	table	table_comment	string	The table comment.
mysql	table	table_create_time	string	When the data file was created.
mysql	table	table_data_free	long	The number of allocated but unused bytes.
mysql	table	table_data_length	long	The length of the data file.
mysql	table	table_engine	string	The storage engine used by a table.

Namespace	Namespace Type	Attribute	Type	Description
mysql	table	table_index_length	long	The length of the index file.
mysql	table	table_max_data_length	long	The maximum length of the data file.
mysql	table	table_name	string	The name of a table.
mysql	table	table_row_format	string	The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact)
mysql	table	table_version	long	The version number of the table's .frm file.
mysql	tables_no_myisam_index_stats	index_stats	string	
mysql	tables_no_unique_keys	key_list	string	
mysql	tablestatus	Auto_increment	long	The next AUTO_INCREMENT value.
mysql	tablestatus	Avg_row_length	long	The average row length.
mysql	tablestatus	Check_time	string	When the table was last checked.
mysql	tablestatus	Checksum	string	The live checksum value (if any)
mysql	tablestatus	Collation	string	The table's character set.
mysql	tablestatus	Comment	string	The comment used when creating the table.
mysql	tablestatus	Create_options	string	Extra options used with CREATE TABLE
mysql	tablestatus	Create_time	string	When the table was created.
mysql	tablestatus	Data_free	long	The number of allocated but unused bytes.
mysql	tablestatus	Data_length	long	The length of the data file.

Namespace	Namespace Type	Attribute	Type	Description
mysql	tablestatus	Engine	string	The storage engine for the table.
mysql	tablestatus	Index_length	long	The length of the index file.
mysql	tablestatus	Max_data_length	long	The maximum length of the data file.
mysql	tablestatus	Name	string	The table name.
mysql	tablestatus	Row_format	string	The row storage format (Fixed, Dynamic, Compressed, Redundant, Compact).
mysql	tablestatus	Rows	long	The number of rows in the table.
mysql	tablestatus	Update_time	string	When the data file was last updated.
mysql	tablestatus	Version	long	The version number of the table's .frm file.
mysql	test_database	Database (test)	string	
mysql	transactions_in_serial_log	transactions_in_serial_log	long	
mysql	trigger_with_select_star	trigger_defn	string	
mysql	user_on_missing_db	db_name	string	
mysql	user_on_missing_db	user	string	
mysql	user_on_missing_table	table	string	
mysql	variables	auto_increment_increment	long	Auto-increment columns are incremented by this value.
mysql	variables	auto_increment_offset	long	Offset added to auto-increment columns.
mysql	variables	autocommit	string	
mysql	variables	automatic_sp_privileges	string	Creating and dropping stored procedures alters ACLs.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>back_log</code>	long	The number of outstanding connection requests MySQL can have.
mysql	variables	<code>basedir</code>	string	General.Directories / Files
mysql	variables	<code>big_tables</code>	string	
mysql	variables	<code>binlog_cache_size</code>	long	Memory.Caches
mysql	variables	<code>binlog_direct_non_transactional_updates</code>	string	
mysql	variables	<code>binlog_format</code>	string	
mysql	variables	<code>bulk_insert_buffer_size</code>	long	Size of tree cache used in bulk insert optimization.
mysql	variables	<code>character_set_client</code>	string	Current client character set.
mysql	variables	<code>character_set_connection</code>	string	Current connection character set.
mysql	variables	<code>character_set_database</code>	string	The character set used by the default database.
mysql	variables	<code>character_set_filesystem</code>	string	Set the file system character set.
mysql	variables	<code>character_set_results</code>	string	Current result character set.
mysql	variables	<code>character_set_server</code>	string	SQL.Charsets
mysql	variables	<code>character_set_system</code>	string	The character set used by the server for storing identifiers.
mysql	variables	<code>character_sets_dir</code>	string	Directory where character sets are.
mysql	variables	<code>collation_connection</code>	string	The collation of the connection.
mysql	variables	<code>collation_database</code>	string	The collation used by the default database.
mysql	variables	<code>collation_server</code>	string	Set the default collation.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>completion_type</code>	long	Default completion type.
mysql	variables	<code>concurrent_insert</code>	long	Use concurrent insert with MyISAM.
mysql	variables	<code>connect_timeout</code>	long	Connections.Overview
mysql	variables	<code>datadir</code>	string	Path to the database root.
mysql	variables	<code>date_format</code>	string	The DATE format (For future)
mysql	variables	<code>datetime_format</code>	string	The DATETIME/ TIMESTAMP format (for future)
mysql	variables	<code>default_week_format</code>	long	The default week format used by WEEK() functions.
mysql	variables	<code>delay_key_write</code>	string	Type of DELAY_KEY_WRITE.
mysql	variables	<code>delayed_insert_limit</code>	long	General.Performance
mysql	variables	<code>delayed_insert_timeout</code>	long	General.Performance
mysql	variables	<code>delayed_queue_size</code>	long	General.Performance
mysql	variables	<code>div_precision_increment</code>	long	SQL.Formats
mysql	variables	<code>engine_condition_pushdown</code>	string	Push supported query conditions to the storage engine.
mysql	variables	<code>error_count</code>	long	
mysql	variables	<code>event_scheduler</code>	string	
mysql	variables	<code>expire_logs_days</code>	long	General.Miscellaneous
mysql	variables	<code>flush</code>	string	Flush tables to disk between SQL commands.
mysql	variables	<code>flush_time</code>	long	A dedicated thread is created to flush all tables at the given interval.
mysql	variables	<code>foreign_key_checks</code>	string	

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>ft_boolean_syntax</code>	string	List of operators for MATCH ... AGAINST (... IN BOOLEAN MODE).
mysql	variables	<code>ft_max_word_len</code>	long	SQL.Full Text Search
mysql	variables	<code>ft_min_word_len</code>	long	SQL.Full Text Search
mysql	variables	<code>ft_query_expansion_limit</code>	long	Number of best matches to use for query expansion.
mysql	variables	<code>ft_stopword_file</code>	string	Use stopwords from this file instead of built-in list.
mysql	variables	<code>general_log</code>	string	
mysql	variables	<code>general_log_file</code>	string	
mysql	variables	<code>group_concat_max_len</code>	long	The maximum length of the result of function group_concat.
mysql	variables	<code>have_archive</code>	string	Whether mysqld supports archive tables.
mysql	variables	<code>have_bdb</code>	string	Is Berkeley DB supported.
mysql	variables	<code>have_blackhole_engine</code>	string	Whether mysqld supports BLACKHOLE tables.
mysql	variables	<code>have_community_features</code>	string	
mysql	variables	<code>have_compress</code>	string	Availability of the zlib compression library.
mysql	variables	<code>have_crypt</code>	string	Availability of the crypt() system call.
mysql	variables	<code>have_csv</code>	string	Whether mysqld supports csv tables.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>have_dynamic_loading</code>	string	Whether mysqld supports dynamic loading of plugins.
mysql	variables	<code>have_example_engine</code>	string	Whether mysqld supports EXAMPLE tables.
mysql	variables	<code>have_federated_engine</code>	string	Whether mysqld supports FEDERATED tables.
mysql	variables	<code>have_geometry</code>	string	Whether mysqld supports spatial data types.
mysql	variables	<code>have_innodb</code>	string	Whether mysqld supports InnoDB tables. No longer used as of MEM 2.3.13.
mysql	variables	<code>have_isam</code>	string	Whether mysqld supports isam tables.
mysql	variables	<code>have_merge_engine</code>	string	Whether mysqld supports merge tables.
mysql	variables	<code>have_ndbcluster</code>	string	Whether mysqld supports NDB Cluster tables.
mysql	variables	<code>have_openssl</code>	string	Whether mysqld supports SSL connections.
mysql	variables	<code>have_partitioning</code>	string	
mysql	variables	<code>have_profiling</code>	string	
mysql	variables	<code>have_query_cache</code>	string	Whether mysqld supports query cache.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	have_raid	string	Whether mysqld supports the RAID option.
mysql	variables	have_rtree_keys	string	General.Features
mysql	variables	have_ssl	string	Whether the server supports an SSL connection.
mysql	variables	have_symlink	string	Is symbolic link support enabled.
mysql	variables	hostname	string	The name of the server host.
mysql	variables	identity	long	
mysql	variables	ignore_builtin_innodb	string	
mysql	variables	init_connect	string	Command(s) that are executed for each new connection.
mysql	variables	init_file	string	Read SQL commands from this file at startup.
mysql	variables	init_slave	string	Command(s) that are executed when a slave connects to a master.
mysql	variables	innodb_adaptive_flushing	string	
mysql	variables	innodb_adaptive_hash_index	string	
mysql	variables	innodb_additional_mem_pool_size	string	Memory.Buffers
mysql	variables	innodb_autoextend_increment	long	Amount by which InnoDB auto-extends the data files for a tablespace, in megabytes.
mysql	variables	innodb_autoinc_lock_mode	long	
mysql	variables	innodb_buffer_pool_awesome_memory	long	Memory.Buffers
mysql	variables	innodb_buffer_pool_size	long	Memory.Buffers
mysql	variables	innodb_change_buffering	string	
mysql	variables	innodb_checksums	string	Enable InnoDB

Namespace	Namespace Type	Attribute	Type	Description
				checksums validation.
mysql	variables	<code>innodb_commit_concurrency</code>	long	Helps in performance tuning in heavily concurrent environments.
mysql	variables	<code>innodb_concurrency_tickets</code>	long	Table Types.InnoDB
mysql	variables	<code>innodb_data_file_path</code>	string	Path to individual files and their sizes.
mysql	variables	<code>innodb_data_home_dir</code>	string	The common location for InnoDB tablespaces
mysql	variables	<code>innodb_doublewrite</code>	string	Enable InnoDB doublewrite buffer
mysql	variables	<code>innodb_fast_shutdown</code>	long	Speeds up the shutdown process of the InnoDB storage engine by deferring cleanup operations until the server restarts.
mysql	variables	<code>innodb_file_format</code>	string	
mysql	variables	<code>innodb_file_format_check</code>	string	
mysql	variables	<code>innodb_file_io_threads</code>	long	Number of file I/O threads in InnoDB.
mysql	variables	<code>innodb_file_per_table</code>	string	Stores each InnoDB table and associated indexes in an .ibd file in the database directory.
mysql	variables	<code>innodb_flush_log_at_trx_commit</code>	string	Table Types.InnoDB
mysql	variables	<code>innodb_flush_method</code>	string	With which method to flush data.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>innodb_force_recovery</code>	long	Table Types.InnoDB
mysql	variables	<code>innodb_io_capacity</code>	long	
mysql	variables	<code>innodb_lock_wait_timeout</code>	long	Table Types.InnoDB
mysql	variables	<code>innodb_locks_unsafe_for_binlog</code>	long	Table Types.InnoDB
mysql	variables	<code>innodb_log_arch_dir</code>	string	Where full logs should be archived.
mysql	variables	<code>innodb_log_archive</code>	string	Set to 1 to have logs archived.
mysql	variables	<code>innodb_log_buffer_size</code>	long	General.Logs
mysql	variables	<code>innodb_log_file_size</code>	long	Size of each log file in a log group.
mysql	variables	<code>innodb_log_files_in_group</code>	long	General.Logs
mysql	variables	<code>innodb_log_group_home_dir</code>	string	Path to InnoDB log files.
mysql	variables	<code>innodb_max_dirty_pages_pct</code>	long	Percentage of dirty pages allowed in InnoDB buffer pool.
mysql	variables	<code>innodb_max_purge_lag</code>	long	Desired maximum length of the purge queue (0 = no limit)
mysql	variables	<code>innodb_mirrored_log_groups</code>	long	Number of identical copies of log groups to keep for the database.
mysql	variables	<code>innodb_old_blocks_pct</code>	long	
mysql	variables	<code>innodb_old_blocks_time</code>	long	
mysql	variables	<code>innodb_open_files</code>	long	The maximum number of files that InnoDB keeps open at the same time.
mysql	variables	<code>innodb_read_ahead_threshold</code>	long	
mysql	variables	<code>innodb_read_io_threads</code>	long	
mysql	variables	<code>innodb_rollback_on_timeout</code>	string	Unknown.Unknown
mysql	variables	<code>innodb_spin_wait_delay</code>	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>innodb_stats_on_metadata</code>	string	
mysql	variables	<code>innodb_stats_sample_pages</code>	long	
mysql	variables	<code>innodb_strict_mode</code>	string	
mysql	variables	<code>innodb_support_xa</code>	string	Enable InnoDB support for the XA two-phase commit.
mysql	variables	<code>innodb_sync_spin_loops</code>	long	Count of spin-loop rounds in InnoDB mutexes
mysql	variables	<code>innodb_table_locks</code>	string	Enable InnoDB locking in LOCK TABLES
mysql	variables	<code>innodb_thread_concurrency</code>	long	Table Types.InnoDB
mysql	variables	<code>innodb_thread_sleep_delay</code>	long	Table Types.InnoDB
mysql	variables	<code>innodb_use_legacy_cardinality_algorithm</code>	string	Applies only to MySQL 5.1 and earlier.
mysql	variables	<code>innodb_use_sys_malloc</code>	string	
mysql	variables	<code>innodb_write_io_threads</code>	long	
mysql	variables	<code>insert_id</code>	long	
mysql	variables	<code>interactive_timeout</code>	long	Connections.Overview
mysql	variables	<code>join_buffer_size</code>	long	The size of the buffer that is used for full joins.
mysql	variables	<code>keep_files_on_create</code>	string	
mysql	variables	<code>key_buffer_size</code>	long	The size of the buffer used for index blocks for MyISAM tables.
mysql	variables	<code>key_cache_age_threshold</code>	long	Memory.Caches
mysql	variables	<code>key_cache_block_size</code>	long	The default size of key cache blocks.
mysql	variables	<code>key_cache_division_limit</code>	long	The minimum percentage of warm blocks in the key cache.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	language	string	Client error messages in given language. May be given as a full path.
mysql	variables	large_files_support	string	Whether large files are supported.
mysql	variables	large_page_size	long	General.Features
mysql	variables	large_pages	string	Enable support for large pages.
mysql	variables	last_insert_id	long	
mysql	variables	lc_time_names	string	General.Miscellaneous
mysql	variables	license	string	Type of license for the server.
mysql	variables	local_infile	string	Enable/disable LOAD DATA LOCAL INFILE (takes values 1 0)
mysql	variables	locked_in_memory	string	Whether mysqld is locked in memory with --memlock.
mysql	variables	log	string	Log connections and queries to file.
mysql	variables	log_bin	string	Enables binary log.
mysql	variables	log_bin_trust_function_creators	string	General.Logs
mysql	variables	log_bin_trust_routine_creators	string	(deprecated) Use log-bin-trust-function-creators.
mysql	variables	log_error	string	Error log file.
mysql	variables	log_output	string	
mysql	variables	log_queries_not_using_indexes	string	General.Logs
mysql	variables	log_slave_updates	string	General.Logs
mysql	variables	log_slow_queries	string	General.Logs
mysql	variables	log_update	string	General.Logs
mysql	variables	log_warnings	long	Log some not critical warnings to the log file.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>long_query_time</code>	double	General.Performance
mysql	variables	<code>low_priority_updates</code>	string	For non-transactional tables, INSERT, DELETE, and UPDATE statements have lower priority than SELECT statements.
mysql	variables	<code>lower_case_file_system</code>	string	General.Features
mysql	variables	<code>lower_case_table_names</code>	long	SQL.Overview
mysql	variables	<code>max_allowed_packet</code>	long	Max packetlength to send/ receive from to server.
mysql	variables	<code>max_binlog_cache_size</code>	long	General.Logs
mysql	variables	<code>max_binlog_size</code>	long	General.Logs
mysql	variables	<code>max_connect_errors</code>	long	Connections.Overview
mysql	variables	<code>max_connections</code>	long	The number of simultaneous clients allowed.
mysql	variables	<code>max_delayed_threads</code>	long	General.Performance
mysql	variables	<code>max_error_count</code>	long	Max number of errors/ warnings to store for a statement.
mysql	variables	<code>max_heap_table_size</code>	long	Don't allow creation of heap tables bigger than this.
mysql	variables	<code>max_insert_delayed_threads</code>	long	This variable is a synonym for <code>max_delayed_threads</code> .
mysql	variables	<code>max_join_size</code>	long	General.Performance
mysql	variables	<code>max_length_for_sort_data</code>	long	Max number of bytes in sorted records.
mysql	variables	<code>max_prepared_stmt_count</code>	long	Maximum number of prepared

Namespace	Namespace Type	Attribute	Type	Description
				statements in the server.
mysql	variables	<code>max_relay_log_size</code>	long	General.Logs
mysql	variables	<code>max_seeks_for_key</code>	long	Limit assumed max number of seeks when looking up rows based on a key.
mysql	variables	<code>max_sort_length</code>	long	The number of bytes to use when sorting BLOB or TEXT values.
mysql	variables	<code>max_sp_recursion_depth</code>	long	Maximum stored procedure recursion depth.
mysql	variables	<code>max_tmp_tables</code>	long	Maximum number of temporary tables a client can keep open at a time.
mysql	variables	<code>max_user_connections</code>	long	Connections.Overview
mysql	variables	<code>max_write_lock_count</code>	long	After this many write locks, allow some read locks to run in between.
mysql	variables	<code>min_examined_row_limit</code>	long	
mysql	variables	<code>multi_range_count</code>	long	General.Performance
mysql	variables	<code>myisam_data_pointer_size</code>	long	Default pointer size to be used for MyISAM tables.
mysql	variables	<code>myisam_max_extra_sort_file_size</code>	long	Deprecated option.
mysql	variables	<code>myisam_max_sort_file_size</code>	long	Table Types.MyISAM
mysql	variables	<code>myisam_mmap_size</code>	long	
mysql	variables	<code>myisam_recover_options</code>	string	The value of the --myisam-recover option.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>myisam_repair_threads</code>	long	Table Types.MyISAM
mysql	variables	<code>myisam_sort_buffer_size</code>	long	Memory.Buffers
mysql	variables	<code>myisam_stats_method</code>	string	Specifies how MyISAM index statistics collection code should treat NULLs.
mysql	variables	<code>myisam_use_mmap</code>	string	
mysql	variables	<code>named_pipe</code>	string	Indicates whether the server supports connections over named pipes.
mysql	variables	<code>net_buffer_length</code>	long	Buffer length for TCP/IP and socket communication.
mysql	variables	<code>net_read_timeout</code>	long	Connections.Networking
mysql	variables	<code>net_retry_count</code>	long	Connections.Networking
mysql	variables	<code>net_write_timeout</code>	long	Connections.Networking
mysql	variables	<code>new</code>	string	Use very new, possibly 'unsafe' functions.
mysql	variables	<code>old</code>	string	
mysql	variables	<code>old_alter_table</code>	string	
mysql	variables	<code>old_passwords</code>	string	Use old password encryption method (needed for 4.0 and older clients)
mysql	variables	<code>open_files_limit</code>	long	General.Directories / Files
mysql	variables	<code>optimizer_prune_level</code>	long	General.Features
mysql	variables	<code>optimizer_search_depth</code>	long	Maximum depth of search performed by the query optimizer.
mysql	variables	<code>optimizer_switch</code>	string	

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>pid_file</code>	string	Pid file used by <code>safe_mysqld</code> .
mysql	variables	<code>plugin_dir</code>	string	
mysql	variables	<code>port</code>	long	Port number to use for connection.
mysql	variables	<code>preload_buffer_size</code>	long	The size of the buffer that is allocated when preloading indexes.
mysql	variables	<code>prepared_stmt_count</code>	long	The current number of prepared statements.
mysql	variables	<code>profiling</code>	string	
mysql	variables	<code>profiling_history_size</code>	string	
mysql	variables	<code>protocol_version</code>	long	The version of the client/server protocol used by the MySQL server.
mysql	variables	<code>pseudo_thread_id</code>	long	
mysql	variables	<code>query_alloc_block_size</code>	long	Allocation block size for query parsing and execution.
mysql	variables	<code>query_cache_limit</code>	long	Don't cache results that are bigger than this.
mysql	variables	<code>query_cache_min_res_unit</code>	long	Memory.Caches
mysql	variables	<code>query_cache_size</code>	long	The memory allocated to store results from old queries.
mysql	variables	<code>query_cache_type</code>	string	Query cache type.
mysql	variables	<code>query_cache_wlock_invalidate</code>	string	Invalidate queries in query cache on LOCK for write.
mysql	variables	<code>query_prealloc_size</code>	long	Persistent buffer

Namespace	Namespace Type	Attribute	Type	Description
				for query parsing and execution.
mysql	variables	rand_seed1	long	
mysql	variables	rand_seed2	long	
mysql	variables	range_alloc_block_size	long	Allocation block size for storing ranges during optimization.
mysql	variables	read_buffer_size	long	Memory.Buffers
mysql	variables	read_only	string	Make all nontemporary tables read only.
mysql	variables	read_rnd_buffer_size	long	Memory.Buffers
mysql	variables	relay_log	string	
mysql	variables	relay_log_index	string	
mysql	variables	relay_log_info_file	string	
mysql	variables	relay_log_purge	string	Determines whether relay logs are purged.
mysql	variables	relay_log_space_limit	long	Maximum space to use for all relay logs.
mysql	variables	report_host	string	
mysql	variables	report_password	string	
mysql	variables	report_port	long	
mysql	variables	report_user	string	
mysql	variables	rpl_recovery_rank	long	Not used.
mysql	variables	secure_auth	string	Disallow authentication for accounts that have old (pre-4.1) passwords.
mysql	variables	secure_file_priv	string	Unknown.Unknown
mysql	variables	server_id	long	General.Overview
mysql	variables	shared_memory	string	Whether the server allows shared-memory connections.
mysql	variables	shared_memory_base_name	string	The name of shared memory

Namespace	Namespace Type	Attribute	Type	Description
				to use for shared-memory connections.
mysql	variables	<code>skip_external_locking</code>	string	Skip system (external) locking.
mysql	variables	<code>skip_name_resolve</code>	string	
mysql	variables	<code>skip_networking</code>	string	Don't allow connection with TCP/IP.
mysql	variables	<code>skip_show_database</code>	string	Don't allow SHOW DATABASE commands.
mysql	variables	<code>slave_compressed_protocol</code>	string	Use compression on master/slave protocol.
mysql	variables	<code>slave_exec_mode</code>	string	
mysql	variables	<code>slave_load_tmpdir</code>	string	General.Replication
mysql	variables	<code>slave_net_timeout</code>	long	Connections.Networking
mysql	variables	<code>slave_skip_errors</code>	string	General.Replication
mysql	variables	<code>slave_transaction_retries</code>	long	General.Replication
mysql	variables	<code>slow_launch_time</code>	long	General.Performance
mysql	variables	<code>slow_query_log</code>	string	
mysql	variables	<code>slow_query_log_file</code>	string	
mysql	variables	<code>socket</code>	string	Socket file to use for connection.
mysql	variables	<code>sort_buffer_size</code>	long	Each thread that needs to do a sort allocates a buffer of this size.
mysql	variables	<code>sql_auto_is_null</code>	string	
mysql	variables	<code>sql_big_selects</code>	string	General.Features
mysql	variables	<code>sql_big_tables</code>	string	
mysql	variables	<code>sql_buffer_result</code>	string	
mysql	variables	<code>sql_log_bin</code>	string	
mysql	variables	<code>sql_log_off</code>	string	
mysql	variables	<code>sql_log_update</code>	string	
mysql	variables	<code>sql_low_priority_updates</code>	string	
mysql	variables	<code>sql_max_join_size</code>	long	

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>sql_mode</code>	string	Set the SQL server mode.
mysql	variables	<code>sql_notes</code>	string	If set to 1, warnings of Note level are recorded.
mysql	variables	<code>sql_quote_show_create</code>	string	
mysql	variables	<code>sql_safe_updates</code>	string	
mysql	variables	<code>sql_select_limit</code>	long	
mysql	variables	<code>sql_slave_skip_counter</code>	long	
mysql	variables	<code>sql_warnings</code>	string	General.Miscellaneous
mysql	variables	<code>ssl_ca</code>	string	CA file in PEM format (check OpenSSL docs, implies --ssl)
mysql	variables	<code>ssl_capath</code>	string	CA directory (check OpenSSL docs, implies --ssl)
mysql	variables	<code>ssl_cert</code>	string	X509 certification in PEM format (implies --ssl)
mysql	variables	<code>ssl_cipher</code>	string	SSL cipher to use (implies --ssl)
mysql	variables	<code>ssl_key</code>	string	X509 key in PEM format (implies ssl)
mysql	variables	<code>storage_engine</code>	string	The default storage engine.
mysql	variables	<code>sync_binlog</code>	long	Synchronously flush binary log to disk after every #th event.
mysql	variables	<code>sync_frm</code>	string	Sync .frm to disk on create. Enabled by default.
mysql	variables	<code>sync_replication</code>	long	Deprecated.
mysql	variables	<code>sync_replication_slave_id</code>	long	Deprecated.
mysql	variables	<code>sync_replication_timeout</code>	long	Deprecated.

Namespace	Namespace Type	Attribute	Type	Description
mysql	variables	<code>system_time_zone</code>	string	The server system time zone.
mysql	variables	<code>table_cache</code>	long	Deprecated; use <code>--table_open_cache</code> instead.
mysql	variables	<code>table_definition_cache</code>	long	
mysql	variables	<code>table_lock_wait_timeout</code>	long	General.Miscellaneous
mysql	variables	<code>table_open_cache</code>	long	
mysql	variables	<code>table_type</code>	string	A synonym for <code>storage_engine</code> .
mysql	variables	<code>thread_cache_size</code>	long	How many threads to keep in a cache for reuse.
mysql	variables	<code>thread_handling</code>	string	
mysql	variables	<code>thread_stack</code>	long	The stack size for each thread.
mysql	variables	<code>time_format</code>	string	The TIME format (for future)
mysql	variables	<code>time_zone</code>	string	The current time zone.
mysql	variables	<code>timed_mutexes</code>	string	General.Miscellaneous
mysql	variables	<code>timestamp</code>	long	
mysql	variables	<code>tmp_table_size</code>	long	Memory.Overview
mysql	variables	<code>tmpdir</code>	string	Path for temporary files.
mysql	variables	<code>transaction_alloc_block_size</code>	long	Allocation block size for transactions to be stored in binary log.
mysql	variables	<code>transaction_prealloc_size</code>	long	Persistent buffer for transactions to be stored in binary log.
mysql	variables	<code>tx_isolation</code>	string	The default transaction isolation level.
mysql	variables	<code>unique_checks</code>	string	
mysql	variables	<code>updatable_views_with_limit</code>	string	SQL.Overview
mysql	variables	<code>version</code>	string	Output version

Namespace	Namespace Type	Attribute	Type	Description
				information and exit.
mysql	variables	<code>version_comment</code>	<code>string</code>	General.Overview
mysql	variables	<code>version_compile_machine</code>	<code>string</code>	The type of machine or architecture on which MySQL was built.
mysql	variables	<code>version_compile_os</code>	<code>string</code>	The type of operating system on which MySQL was built.
mysql	variables	<code>wait_timeout</code>	<code>long</code>	Connections.Overview
mysql	variables	<code>warning_count</code>	<code>long</code>	
mysql	waiting_db	<code>waiting_db</code>	<code>string</code>	
mysql	waiting_statement	<code>waiting_statement</code>	<code>string</code>	
mysql	waiting_thread	<code>waiting_thread</code>	<code>long</code>	
mysql	waiting_time	<code>waiting_time</code>	<code>long</code>	
mysql	waiting_user	<code>waiting_user</code>	<code>string</code>	
os	cpu	<code>cpu_cache_size</code>	<code>long</code>	The size of the CPU cache.
os	cpu	<code>cpu_idle</code>	<code>long_counter</code>	The CPU idle time.
os	cpu	<code>cpu_mhz</code>	<code>long</code>	The MHz rating of the CPU.
os	cpu	<code>cpu_name</code>	<code>string</code>	The CPU name.
os	cpu	<code>cpu_sys</code>	<code>long_counter</code>	CPU system / kernel time.
os	cpu	<code>cpu_user</code>	<code>long_counter</code>	CPU user space time.
os	cpu	<code>cpu_vendor</code>	<code>string</code>	The name of the CPU vendor.
os	cpu	<code>cpu_wait</code>	<code>long_counter</code>	CPU IO wait time.
os	disk	<code>disk_bytes_read</code>	<code>long_counter</code>	The number of bytes read from disk.
os	disk	<code>disk_bytes_written</code>	<code>long_counter</code>	The number of bytes written to disk.

Namespace	Namespace Type	Attribute	Type	Description
os	disk	disk_queue	long	The size of the disk IO queue.
os	disk	disk_reads	long_counter	The number of disk reads.
os	disk	disk_rtime	long	
os	disk	disk_servicetime	double	
os	disk	disk_snaptime	long	
os	disk	disk_time	long	
os	disk	disk_writes	long_counter	The number of disk writes.
os	disk	disk_wtime	long	
os	fs	fs_avail	long	
os	fs	fs_dev_name	string	
os	fs	fs_dir_name	string	
os	fs	fs_files	long	
os	fs	fs_flags	long	
os	fs	fs_free_files	long	
os	fs	fs_options	string	
os	fs	fs_sys_type_name	string	
os	fs	fs_total	long	
os	fs	fs_type_name	string	
os	fs	fs_unused	long	
os	fs	fs_used	long	
os	Host	host_id	string	The host unique identifier.
os	Host	name	string	
os	loadavg	0	double	
os	loadavg	1	double	
os	loadavg	2	double	
os	mem	ram_total	long	The total amount of available RAM.
os	mem	ram_unused	long	The total amount of unused RAM.
os	mem	swap_page_in	long_counter	
os	mem	swap_page_out	long_counter	
os	mem	swap_total	long	The total amount of available swap memory.

Namespace	Namespace Type	Attribute	Type	Description
os	mem	swap_unused	long	The total amount of unused swap memory.
os	net	address	string	The address of the network interface.
os	net	broadcast	string	The network interface broadcast address.
os	net	description	string	The network interface description.
os	net	destination	string	The network interface destination address.
os	net	flags	long	The network interface flags.
os	net	hwaddr	string	The network interface MAC address.
os	net	metric	long	Operating System.Net
os	net	mtu	long	The Maximum Transmission Unit.
os	net	name	string	The network interface name (eth0, eth1 etc.)
os	net	netmask	string	The network interface subnet mask.
os	net	rx_bytes	long_counter	The number of bytes received.
os	net	rx_dropped	long_counter	The number of received packets dropped.
os	net	rx_errors	long_counter	The number of received packet errors.
os	net	rx_frame	long_counter	The number received packet frame errors.

Namespace	Namespace Type	Attribute	Type	Description
os	net	<code>rx_overruns</code>	<code>long_counter</code>	The number of received packet overruns.
os	net	<code>rx_packets</code>	<code>long_counter</code>	The number of packets received.
os	net	<code>speed</code>	<code>long</code>	The network interface connection speed.
os	net	<code>tx_bytes</code>	<code>long_counter</code>	The number of bytes transmitted.
os	net	<code>tx_carrier</code>	<code>long_counter</code>	The number of carrier errors on network transmission.
os	net	<code>tx_collisions</code>	<code>long_counter</code>	The number of network transmission collisions.
os	net	<code>tx_dropped</code>	<code>long_counter</code>	The number of network transmissions dropped.
os	net	<code>tx_errors</code>	<code>long_counter</code>	The number of network transmission errors.
os	net	<code>tx_overruns</code>	<code>long_counter</code>	The number of network transmission overruns.
os	net	<code>tx_packets</code>	<code>long_counter</code>	The number of packets transmitted.
os	net	<code>type</code>	<code>string</code>	The network interface type (ethernet, loopback etc.)
os	os	<code>os_arch</code>	<code>string</code>	The architecture of the OS / CPU.
os	os	<code>os_description</code>	<code>string</code>	Description of the OS.
os	os	<code>os_machine</code>	<code>string</code>	Machine CPU architecture.
os	os	<code>os_name</code>	<code>string</code>	The OS name.

Namespace	Namespace Type	Attribute	Type	Description
os	os	os_patchlevel	string	The OS patch level.
os	os	os_vendor	string	The OS vendor.
os	os	os_vendor_code_name	string	The OS vendor code name.
os	os	os_vendor_name	string	The OS vendor name.
os	os	os_vendor_version	string	The OS vendor version number.
os	os	os_version	string	The OS version number.
util	LogHistogram	0	long_counter	
util	LogHistogram	1	long_counter	
util	LogHistogram	10	long_counter	
util	LogHistogram	11	long_counter	
util	LogHistogram	12	long_counter	
util	LogHistogram	13	long_counter	
util	LogHistogram	14	long_counter	
util	LogHistogram	15	long_counter	
util	LogHistogram	16	long_counter	
util	LogHistogram	17	long_counter	
util	LogHistogram	18	long_counter	
util	LogHistogram	19	long_counter	
util	LogHistogram	2	long_counter	
util	LogHistogram	20	long_counter	
util	LogHistogram	21	long_counter	
util	LogHistogram	22	long_counter	
util	LogHistogram	23	long_counter	
util	LogHistogram	24	long_counter	
util	LogHistogram	25	long_counter	
util	LogHistogram	26	long_counter	
util	LogHistogram	27	long_counter	
util	LogHistogram	28	long_counter	
util	LogHistogram	29	long_counter	
util	LogHistogram	3	long_counter	
util	LogHistogram	30	long_counter	
util	LogHistogram	31	long_counter	
util	LogHistogram	32	long_counter	
util	LogHistogram	33	long_counter	

Namespace	Namespace Type	Attribute	Type	Description
util	LogHistogram	34	long_counter	
util	LogHistogram	35	long_counter	
util	LogHistogram	36	long_counter	
util	LogHistogram	37	long_counter	
util	LogHistogram	38	long_counter	
util	LogHistogram	39	long_counter	
util	LogHistogram	4	long_counter	
util	LogHistogram	40	long_counter	
util	LogHistogram	41	long_counter	
util	LogHistogram	42	long_counter	
util	LogHistogram	43	long_counter	
util	LogHistogram	44	long_counter	
util	LogHistogram	45	long_counter	
util	LogHistogram	46	long_counter	
util	LogHistogram	47	long_counter	
util	LogHistogram	48	long_counter	
util	LogHistogram	49	long_counter	
util	LogHistogram	5	long_counter	
util	LogHistogram	50	long_counter	
util	LogHistogram	51	long_counter	
util	LogHistogram	52	long_counter	
util	LogHistogram	53	long_counter	
util	LogHistogram	54	long_counter	
util	LogHistogram	55	long_counter	
util	LogHistogram	56	long_counter	
util	LogHistogram	57	long_counter	
util	LogHistogram	58	long_counter	
util	LogHistogram	59	long_counter	
util	LogHistogram	6	long_counter	
util	LogHistogram	60	long_counter	
util	LogHistogram	61	long_counter	
util	LogHistogram	62	long_counter	
util	LogHistogram	7	long_counter	
util	LogHistogram	8	long_counter	
util	LogHistogram	9	long_counter	
util	LogHistogram	base	long	
util	LogHistogram	nul	long_counter	

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](#).

B

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

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](#).

M

master

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](#).

master/slave

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](#).

metric

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](#).

mixed replication

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](#).

monitor

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](#).

My Oracle Support

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](#).

MySQL server

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](#).

N

.NET

A programming framework used by many Windows applications, including the **Connector/Net** component.

See Also [Connector/Net](#).

NMS

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](#).

normalized query

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 `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 `SELECT` queries, but also DML statements such as `INSERT`, `UPDATE`, and `DELETE`, and DDL statements such as `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](#).

R

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](#).

S

schedule

Determines 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

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 Closed, 202

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
- checkmysqlhost 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
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
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
plugin, 375
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
- repository, 376
 - database name, 82
- restartImmediately option, 83
- restore
 - backup, 69
- restoring
 - backup, 69
- role, 376
- rule, 376
- rules
 - blackout periods, 244
 - creating, 232
 - variable substitution, 235

S

- 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 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

