

MySQL Enterprise Monitor 3.1.4 Manual

Abstract

This manual documents the MySQL Enterprise Monitor version 3.1.4.

For notes detailing the changes in each release, see the [MySQL Enterprise Monitor 3.1 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: 2016-06-04 (revision: 6489)

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	10
2 What's New in 3.1	11
2.1 Security	11
2.2 Performance Tuning	11
2.3 Usability	12
II Installing MySQL Enterprise Monitor	13
3 Installation Prerequisites	17
3.1 Installer Files	17
3.2 Prerequisites	18
3.2.1 System Requirements	18
3.2.2 Supported Platforms	19
3.2.3 MySQL Enterprise Monitor Repository	21
3.3 Credentials Required for Installation	22
3.3.1 Existing Users	22
3.3.2 Users Created During Installation	22
3.3.3 Users Created on First Log-in	23
3.4 Supported Browsers	23
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	31
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 Monitoring Agent Graphical Installation Wizard	37
5.5 Starting/Stopping the MySQL Enterprise Monitor Agent	40
5.5.1 Starting/Stopping the Agent on Windows	40
5.5.2 Starting/Stopping the Agent on Mac OS X	42
5.5.3 Starting/Stopping the Agent on Unix	43
5.5.4 sql_mode	44
5.6 Monitoring Multiple MySQL Servers	44
5.7 Configuring an Agent to Monitor a Remote MySQL Server	44
5.8 Monitoring Outside the Firewall with an SSH Tunnel	45
5.9 HTTP Connection Timeout	46
5.10 Troubleshooting the Agent	46
5.11 Agent Backlog	48
6 Upgrading MySQL Enterprise Monitor Installations	49
6.1 General considerations when upgrading MySQL Enterprise Monitor	49
6.2 Upgrading to MySQL Enterprise Monitor 3.1.x	49
6.3 Restoring from Backup	51
7 Post-installation Considerations	53
7.1 General Considerations	53

7.2 Installing SSL Certificates	54
7.3 Changing an SSH Host Key	57
8 Unattended Installation Reference	59
8.1 Unattended Installation	59
8.1.1 Performing an Unattended Installation	59
8.1.2 MySQL Enterprise Service Manager Options	60
8.1.3 MySQL Enterprise Monitor Agent Options	66
9 Performance Tuning MySQL Enterprise Monitor	75
9.1 Tuning Memory	75
9.2 Tuning CPU	77
9.3 Tuning Apache Tomcat Threads	78
9.4 Tuning Agent Memory Requirements	79
10 Configuration Utilities	81
10.1 Service Manager Configuration Utilities	81
10.2 Agent Configuration Utility	83
11 Proxy and Aggregator Installation	87
11.1 Proxy Aggregator Architecture	87
11.2 Prerequisites	88
11.3 Installing the Proxy and Aggregator	89
11.4 Graphical Installation Wizard	89
11.5 Text-Based Installation	91
11.6 Unattended Installation	92
11.7 Starting and Stopping the Proxy and Aggregator	95
11.8 Configuration Options	96
12 Configuring Connectors	101
12.1 Using the MySQL Enterprise Plugin for Connector/PHP	101
12.2 Using the MySQL Enterprise Plugin for Connector/J	105
12.3 Using the MySQL Enterprise Plugin for Connector/Net	110
III Using MySQL Enterprise Monitor	113
13 User Interface	117
13.1 Initial Log-In	117
13.2 Setting the Timezone and Locale	118
13.3 Menus and Toolbars	118
13.3.1 Main Menus	118
13.3.2 Status Summary	120
14 Overview	123
14.1 Database Statistics	123
14.2 Overview Graphs	124
14.3 General Database Statistics	125
14.4 Group Overview Configuration	125
15 MySQL Instances Dashboard	127
15.1 MySQL Instance Dashboard UI	127
15.2 MySQL Instance Details	129
15.3 Adding Instances	131
15.3.1 Adding a MySQL Instance	131
15.3.2 Adding Multiple MySQL Instances	135
15.4 Filtering MySQL Instances	135
16 Managing Groups of Instances	137
17 Replication Dashboard	139
18 Reports and Graphs	143
18.1 All Timeseries Graphs	143
18.1.1 Graph Controls	143
18.1.2 Graph Types	145
18.2 Database File I/O and Lock Waits	145

18.2.1	sys Schema	145
18.2.2	Database File I/O Graphs and Reports	146
18.2.3	Lock Waits Report	148
18.3	InnoDB Buffer Pool Usage	148
19	Advisors	151
19.1	Advisors Page	151
19.2	Advisor Types	155
19.3	Advisor Thresholds	156
19.4	Advisor Schedules	158
20	Events and Event Handling	159
20.1	Events	159
20.2	Event Handling	163
20.2.1	Event Handlers	163
20.2.2	Event Handling Page	163
20.3	Creating Event Handlers	168
20.3.1	Event Action Log	170
20.3.2	Suspending an Event Handler	170
21	Expression-Based Advisor Reference	171
21.1	Administration Advisors	171
21.2	Agent Advisors	177
21.3	Availability Advisors	178
21.4	Cluster Advisors	179
21.5	Memory Usage Advisors	181
21.6	Monitoring and Support Services Advisors	183
21.7	Operating System Advisors	184
21.8	Performance Advisors	184
21.9	Replication Advisors	190
21.10	Schema Advisors	195
21.11	Security Advisors	199
22	GUI-Based Advisor Reference	207
22.1	Agent Health Advisor	207
22.2	MySQL Enterprise Backup Health Advisor	210
22.3	MySQL Process Discovery Advisor	211
22.4	Duplicate MySQL Server UUID	212
22.5	HTTP Server KeyStore's Certificate About to Expire	212
22.6	sys Schema Install Advisor	212
22.7	CPU Utilization Advisor	213
22.8	Filesystem Free Space Advisor	214
22.9	MySQL Process	216
22.10	Query Analysis Advisors	216
22.11	Security Advisors	217
23	Access Control	219
23.1	Users and Roles	219
23.2	Permissions	220
23.3	Monitored Assets Permissions	220
23.3.1	Server Group	221
23.3.2	MySQL Instances	221
23.4	Monitoring Services	223
23.5	MySQL Enterprise Monitor	223
23.6	Default Users and Roles	225
23.7	Creating Users and Roles	226
24	Access Control - Best Practices	229
24.1	Open Permission Sets	230
24.2	Strict Permission Set	232

25 Global Settings	237
25.1 Server Locale	237
25.2 Server Hostname	237
25.3 Customize MySQL Server Name	237
25.4 Data Purge Behavior	239
25.5 My Oracle Support Credentials	239
25.6 HTTP Proxy Settings	240
25.7 External Authentication	240
26 Customizing MySQL Enterprise Monitor	245
26.1 Creating Advisors and Rules	245
26.1.1 Creating Advisors	245
26.1.2 Overview of Graph Creation	246
26.1.3 Overview of Advisor Creation	247
26.1.4 Variables	248
26.1.5 Thresholds	248
26.1.6 Using Strings	249
26.1.7 Wiki Format	250
26.1.8 Creating a New Advisor: An Example	250
26.1.9 Creating a New Graph: An Example	251
26.2 Custom Data Collection	252
26.2.1 Custom.xml	253
26.2.2 Queries	253
26.2.3 Data Collection Attributes	255
26.3 Event Notification Blackout Periods	256
26.3.1 Scripting Blackouts	257
IV Using the Query Analyzer	259
27 Using the Query Analyzer	263
27.1 Providing Query Analyzer Data	264
27.1.1 Using the MySQL Performance Schema	264
27.2 Query Response Time index (QRTi)	267
27.3 Query Analyzer User Interface	268
27.3.1 Getting Detailed Query Information	270
27.3.2 Using Graphs to Identify Queries	271
27.3.3 Filtering Query Analyzer Data	272
27.3.4 Query Analyzer Settings	273
27.3.5 Exporting Query Information	274
V Appendices	275
A MySQL Enterprise Monitor Component Reference	279
A.1 MySQL Enterprise Service Manager Reference	279
A.1.1 Log Files for the MySQL Enterprise Service Manager	279
A.1.2 The Management Information Base (MIB) File	279
A.1.3 The <code>config.properties</code> file	279
A.2 MySQL Enterprise Monitor Agent Reference	284
A.2.1 Agent Log Files	284
B Managing the Inventory	287
B.1 The Inventory Page	287
B.2 Using the Inventory Page	288
C MySQL Enterprise Monitor Frequently Asked Questions	289
D MySQL Enterprise Monitor Support	295
D.1 Diagnostics Report	295
Index	297

List of Figures

1.1 MySQL Enterprise Monitor Architecture	5
1.2 MySQL Enterprise Monitor Agentless Architecture	6
11.1 MySQL Enterprise Monitor Proxy and Aggregator Architecture	88
12.1 Plugin for PHP and Aggregator Architecture	101
12.2 Connector Plugin Architecture	106
12.3 Connector Plugin Architecture	110
13.1 Initial setup for the MySQL Enterprise Monitor User Interface	117
13.2 Status Summary	120
14.1 Overview Dashboard	123
14.2 Group Overview Filter Configuration	125
15.1 Add Instance Connection Settings	132
15.2 Add Instance Encryption Settings	133
15.3 Add Instance Advanced Settings	134
15.4 MySQL Instance Filter	136
16.1 Group Management Page	137
18.1 Database File I/O By File	146
18.2 Database File I/O By Wait Type Report	147
18.3 Database File I/O By Wait Type Graphs	147
18.4 Database File I/O By Thread	148
19.1 Advisors Page	151
19.2 Advisor Menu Control	153
19.3 Advisor Pop-up Menu	153
19.4 Advisor Pop-up Menu	155
19.5 Agent Health - General	156
19.6 Threshold Definitions Example	157
20.1 Events Page with Filter	160
20.2 Event Handlers section	164
20.3 Email Notification Groups section	165
20.4 Create Group Dialog	166
20.5 Email Settings section	166
20.6 SNMP Settings section	167
22.1 Agent Health - General	208
22.2 Agent Health - General	209
22.3 Agent Health - Backlog	210
22.4 CPU Usage	213
22.5 CPU Outliers	214
22.6 Filesystem - General	214
22.7 Filesystem - Estimated Full Capacity	215
22.8 Filesystem - Percentage of Space	215
22.9 Filesystem - Percentage Used in Time Range	216
23.1 Core Monitored Assets	221
23.2 Monitoring Services Permissions	223
24.1 Hybrid Permission Set Overview	232
24.2 Strict Permission Set Grouped	236
25.1 Server Hostname	237
25.2 Customize MySQL Server Name	238
25.3 Data Purge Behavior	239
25.4 My Oracle Support Credentials	240
25.5 HTTP Proxy Settings	240
25.6 External Authentication Settings: LDAP	241
27.1 MySQL Enterprise Monitor User Interface: Query Analyzer	268

List of Tables

3.1 Disk space Required	18
4.1 Installation Parameters	26
7.1 SSL Configuration Options For The Agent's <code>bootstrap.properties</code>	56
8.1 MySQL Enterprise Service Manager Installer Options	60
8.2 MySQL Enterprise Monitor Agent Installer Options	67
9.1 Apache Tomcat configuration file location (default)	75
9.2 Installation Parameters	75
9.3 MEM repository configuration file location (default)	76
9.4 MEM repository configuration tool location (default)	77
9.5 MEM repository configuration tool location (default)	78
10.1 MEM Repository Configuration Tool Location (default)	81
10.2 Service Manager Config Utilities	81
10.3 Service Manager Certificate Utilities	83
10.4 MEM Agent Configuration Tool Location (default)	83
10.5 Agent Connection Utility	84
10.6 Agent Configuration Utility	85
11.1 MySQL Enterprise Monitor Proxy and Aggregator Installer Options	92
11.2 Proxy and Aggregator Help Options	96
11.3 Application Options	96
11.4 aggr-module Options	97
11.5 proxy-module Options	98
12.1 Connector/PHP Properties	104
12.2 MySQL Plugin for Connector/J Properties	107
12.3 MySQL Plugin for Connector/J SSL Properties	108
14.1 Group Overview Filter Configuration	125
15.1 Bad Connection List	128
15.2 Unreachable Agents List	128
15.3 Unmonitored MySQL Instances List	129
15.4 MySQL Instance Details Columns	130
15.5 Connection Settings Tab	132
15.6 Encryption Settings Tab	134
15.7 Advanced Settings	134
15.8 MySQL Instance Filter	136
18.1 Timeseries Graph Filter	144
19.1 Advisor Page Controls	152
19.2 Advisor Information Listing	153
19.3 Advisor Edit Menu Controls	154
19.4 Advisor Filter Controls	155
20.1 Events Filter Controls	160
20.2 Events List Columns	161
20.3 Event Handler List Controls	164
20.4 Email Notification Groups Controls	165
20.5 Email Settings Controls	167
20.6 SNMP Settings Controls	167
20.7 Create Event Handler Controls	168
22.1 MySQL Process Discovery Controls	211
24.1 Manager Role Definition	230
24.2 DBA Role Definition	231
24.3 System-Wide Role Definition	233
24.4 Development Group Role Definition	234
24.5 Production Group Role Definition	235

25.1 Customize	237
25.2 Customize	238
25.3 External Authentication	240
25.4 LDAP Authentication	241
25.5 Active Directory Authentication	243
26.1 MySQL Enterprise Monitor: Wiki Formatting	250
26.2 Custom Data Collection Class Elements	253
26.3 Attribute Elements	255
27.1 QRTi value definitions	267
A.1 MySQL Enterprise Monitor: Log File Locations	279
A.2 MySQL Enterprise Monitor: MIB File Locations	279
A.3 MySQL Enterprise Monitor: Default path of the config.properties File	280
A.4 Optional config.properties values	280

Preface and Legal Notices

This manual documents the MySQL Enterprise Monitor version 3.1.4.

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, 2016, 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.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

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	10
2 What's New in 3.1	11
2.1 Security	11
2.2 Performance Tuning	11
2.3 Usability	12

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	10



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/>.

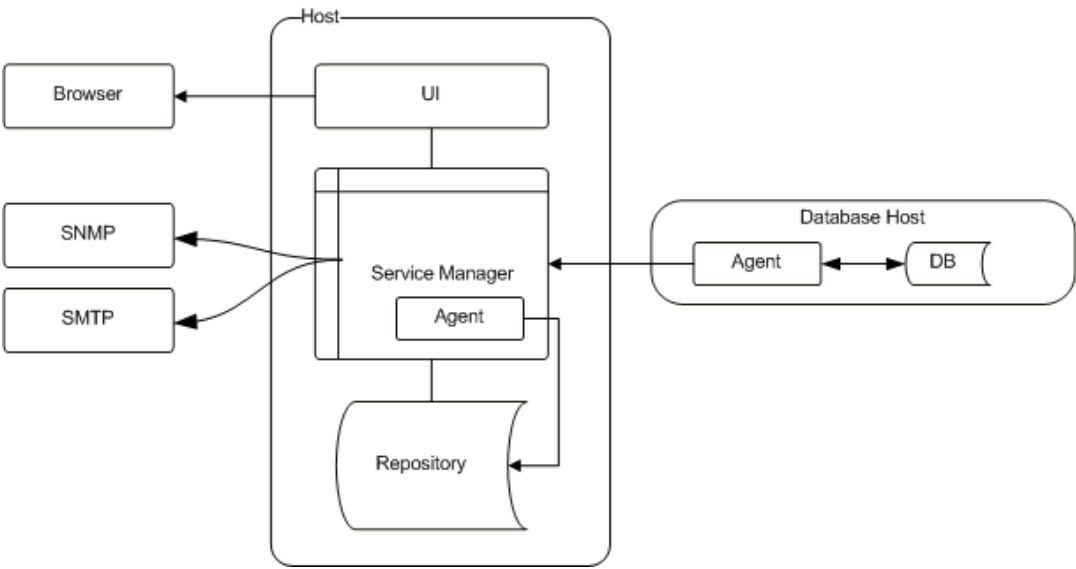
MySQL Enterprise Monitor is a companion product to MySQL Server that enables monitoring of MySQL instances and their hosts, notification of potential issues and problems, and advice on how to correct issues. MySQL Enterprise Monitor can monitor all types of installation, from a single MySQL instance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

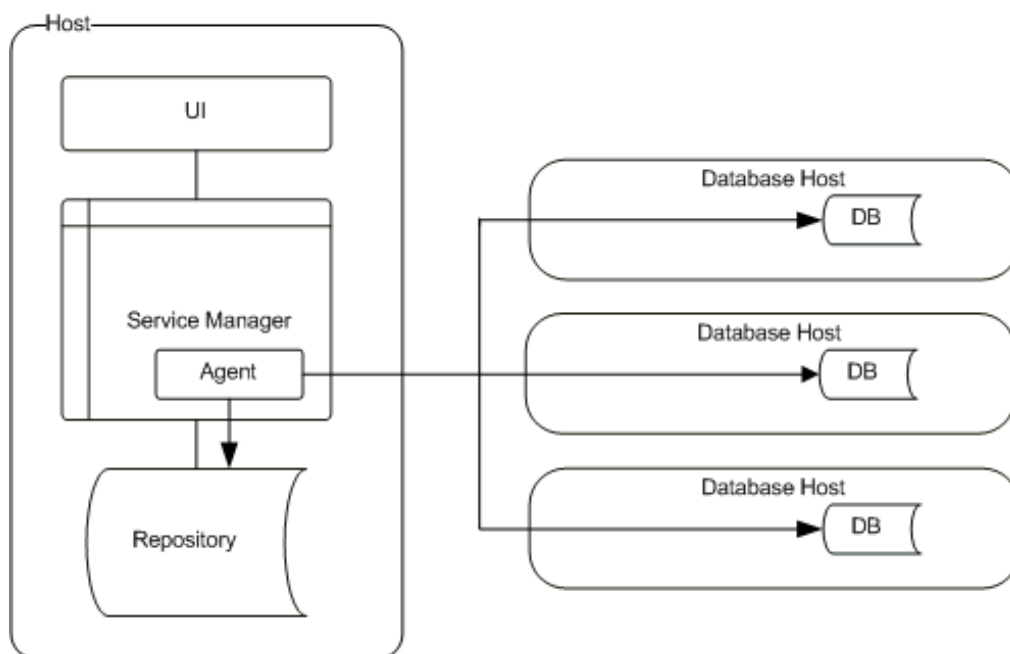


MySQL Enterprise Monitor has the following components:

- 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



Important

For performance reasons, the agentless installation is not recommended for monitoring large implementations. It is useful for testing, or very small-scale implementations. It is strongly recommended to install an agent on each host.

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.
- **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 3.1

Table of Contents

2.1 Security	11
2.2 Performance Tuning	11
2.3 Usability	12

This section provides a high-level overview of the differences between this release and its predecessor.

2.1 Security

Access Control Lists

MySQL Enterprise Monitor 3.1 introduces Access Control List (ACL) support. MySQL Enterprise Monitor ACL enables the following:

- Define visibility: strictly limit access to specific groups of assets or grant access to all assets.
- Define Roles: rather than define permissions per user, as in previous releases, permission sets are defined in roles and multiple users can be assigned to each role. It is also possible to assign users to multiple different roles.
- Restrict access to sensitive data: grant or deny rights to view specific types of potentially sensitive data, such as Query Analyzer data.
- Authenticate using external services: map your users to roles defined in LDAP or Active Directory.

For more information, see [Chapter 23, Access Control](#) and [Chapter 24, Access Control - Best Practices](#).

MySQL Enterprise Firewall Monitoring

The MySQL Enterprise Firewall plugin hardens MySQL against threats such as SQL injection or attempts to exploit applications by using them outside of their legitimate query workload characteristics. MEM 3.1 notifies you if a potential security threat has been identified and provides pro-active configuration advice if your settings can be improved. View graphs of Access Denied, Access Granted and Suspicious Access counters over time for the MySQL instances you wish to safeguard. For more information on MySQL Enterprise Firewall, see [MySQL Enterprise Firewall](#).

MySQL Enterprise Audit Monitoring

Monitor and enforce MySQL Enterprise Audit configuration and usage across all of your MySQL servers. MEM 3.1's policy-based Advisor and event-handling track and inform you about MySQL Enterprise Audit status (enabled/disabled), write waits, and lost events to help you ensure best practices and regulatory compliance. For more information on MySQL Enterprise Audit, see [MySQL Enterprise Audit](#).

2.2 Performance Tuning

Identify I/O hot spots and lock wait contention in your application using the new Database File I/O and Lock Waits reports. These reports utilize `sys` schema which can be installed on the selected instance from within MySQL Enterprise Monitor. For more information, see [Section 18.2, "Database File I/O and Lock Waits"](#).



Note

`sys` schema is supported on MySQL 5.6 and higher, only.

2.3 Usability

The following changes were made to the user interface.

- New and improved Group Editor available from the Settings menu.
- New User and Roles editor available from the Settings menu.
- Customizable graphs on the **Overview** dashboard.
- New File I/O and Lock Waits reports and graphs available from the **Reports & Graphs** page.

Part II Installing MySQL Enterprise Monitor

Table of Contents

3	Installation Prerequisites	17
3.1	Installer Files	17
3.2	Prerequisites	18
3.2.1	System Requirements	18
3.2.2	Supported Platforms	19
3.2.3	MySQL Enterprise Monitor Repository	21
3.3	Credentials Required for Installation	22
3.3.1	Existing Users	22
3.3.2	Users Created During Installation	22
3.3.3	Users Created on First Log-in	23
3.4	Supported Browsers	23
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	31
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	Monitoring Agent Graphical Installation Wizard	37
5.5	Starting/Stopping the MySQL Enterprise Monitor Agent	40
5.5.1	Starting/Stopping the Agent on Windows	40
5.5.2	Starting/Stopping the Agent on Mac OS X	42
5.5.3	Starting/Stopping the Agent on Unix	43
5.5.4	sql_mode	44
5.6	Monitoring Multiple MySQL Servers	44
5.7	Configuring an Agent to Monitor a Remote MySQL Server	44
5.8	Monitoring Outside the Firewall with an SSH Tunnel	45
5.9	HTTP Connection Timeout	46
5.10	Troubleshooting the Agent	46
5.11	Agent Backlog	48
6	Upgrading MySQL Enterprise Monitor Installations	49
6.1	General considerations when upgrading MySQL Enterprise Monitor	49
6.2	Upgrading to MySQL Enterprise Monitor 3.1.x	49
6.3	Restoring from Backup	51
7	Post-installation Considerations	53
7.1	General Considerations	53
7.2	Installing SSL Certificates	54
7.3	Changing an SSH Host Key	57
8	Unattended Installation Reference	59
8.1	Unattended Installation	59
8.1.1	Performing an Unattended Installation	59
8.1.2	MySQL Enterprise Service Manager Options	60
8.1.3	MySQL Enterprise Monitor Agent Options	66
9	Performance Tuning MySQL Enterprise Monitor	75
9.1	Tuning Memory	75
9.2	Tuning CPU	77
9.3	Tuning Apache Tomcat Threads	78
9.4	Tuning Agent Memory Requirements	79

10 Configuration Utilities	81
10.1 Service Manager Configuration Utilities	81
10.2 Agent Configuration Utility	83
11 Proxy and Aggregator Installation	87
11.1 Proxy Aggregator Architecture	87
11.2 Prerequisites	88
11.3 Installing the Proxy and Aggregator	89
11.4 Graphical Installation Wizard	89
11.5 Text-Based Installation	91
11.6 Unattended Installation	92
11.7 Starting and Stopping the Proxy and Aggregator	95
11.8 Configuration Options	96
12 Configuring Connectors	101
12.1 Using the MySQL Enterprise Plugin for Connector/PHP	101
12.2 Using the MySQL Enterprise Plugin for Connector/J	105
12.3 Using the MySQL Enterprise Plugin for Connector/Net	110

Chapter 3 Installation Prerequisites

Table of Contents

3.1 Installer Files	17
3.2 Prerequisites	18
3.2.1 System Requirements	18
3.2.2 Supported Platforms	19
3.2.3 MySQL Enterprise Monitor Repository	21
3.3 Credentials Required for Installation	22
3.3.1 Existing Users	22
3.3.2 Users Created During Installation	22
3.3.3 Users Created on First Log-in	23
3.4 Supported Browsers	23

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. 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, 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

Depending on your configuration and environment, download these files from your [My Oracle Support \(MOS\)](#) page. Or if you are interested in a trial, then you may use the [Oracle Software Delivery Cloud \(OSDC\)](#). 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.extension`. For an upgrade installation, this installer is named `mysqlmonitor-version-platform-update-installer.extension`.

- 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	N/A	600 MB
Linux x86 64-bit	1.3 GB	800 MB
Mac OS X	1.2 GB	700 MB
Solaris x86 32-bit	N/A	600 MB
Solaris x86 64-bit	1.8 GB	800 MB

Platform	Minimum Disk space Required by Service Manager	Minimum Disk space Required by Monitoring Agent
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	N/A	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.
- The MySQL Enterprise Service Manager installation generates a self-signed certificate during the installation process. This certificate generation requires a valid, resolvable hostname. If the host

on which you install the MySQL Enterprise Service Manager does not have a valid hostname, the installation will fail.



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.

MySQL Requirements

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

- The Enterprise Server Manager repository requires MySQL Server 5.6.14 or higher. 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.

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.



Note

It is possible to use MySQL 5.5.8, or 5.6.10, or higher, but 5.6.14 is the recommended minimum version for the MySQL Enterprise Monitor repository.

- The InnoDB storage engine is available.
- SSL is enabled.

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:
 - `CREATE`
 - `ALTER`
 - `DROP`
 - `INDEX`
 - `CREATE TEMPORARY TABLES`
 - `CREATE USER`

- [DELETE](#)
- [INSERT](#)
- [PROCESS](#)
- [REPLICATION CLIENT](#)
- [SELECT](#)
- [SHOW DATABASES](#)
- [UPDATE](#)

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

3.3 Credentials Required 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”](#).

3.4 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

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	31

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 3.1.0 cannot communicate with any MySQL Enterprise Monitor Agent earlier than version 3.0.22. It is strongly recommended you use 3.1 agents with the 3.1 Service Manager.

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.
- 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	512MB	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`.

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 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 3.1, 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 3.1, 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 4.2, “MySQL Enterprise Service Manager 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.

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 Monitoring Agent Graphical Installation Wizard	37
5.5 Starting/Stopping the MySQL Enterprise Monitor Agent	40
5.5.1 Starting/Stopping the Agent on Windows	40
5.5.2 Starting/Stopping the Agent on Mac OS X	42
5.5.3 Starting/Stopping the Agent on Unix	43
5.5.4 sql_mode	44
5.6 Monitoring Multiple MySQL Servers	44
5.7 Configuring an Agent to Monitor a Remote MySQL Server	44
5.8 Monitoring Outside the Firewall with an SSH Tunnel	45
5.9 HTTP Connection Timeout	46
5.10 Troubleshooting the Agent	46
5.11 Agent Backlog	48

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.

It is strongly recommended that MySQL Enterprise Monitor Agent 3.1 are installed for use with the MySQL Enterprise Service Manager 3.1.

5.1 General Agent Related Notes

This section describes important features of the 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.

- Proxy and Aggregator



Note

The Proxy and Aggregator are not included in the MySQL Enterprise Monitor Agent 3.1.0 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](#)

- You can 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-agent.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.

- **Admin:** a user that has the `SUPER`, `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 intend to automatically create the less-privileged users, General and Limited, you must also grant the Admin user `CREATE USER`, `SHOW VIEW`, `PROCESS`, `REPLICATION CLIENT`, `SELECT` and `SHOW`

DATABASES privileges globally, and `UPDATE` on the `performance_schema.threads` table, `WITH GRANT OPTION` for all..

If you intend to install the sys schema from within MySQL Enterprise Monitor, in addition to the privileges listed above, you must also grant the Admin user `CREATE ROUTINE`, `CREATE TEMPORARY TABLES`, `CREATE VIEW`, and `TRIGGER`.

- **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 VIEWS`.



Important

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 VIEWS`.

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* 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 5.7, “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 Monitoring Agent Graphical Installation Wizard

This section describes how to install the Agent using the Installation Wizard. The steps are identical in the command line installation method.

**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.

You can also install the Monitor Agent in `unattended` mode. For more information on unattended installation, see [Section 8.1, “Unattended Installation”](#).

**Note**

To install multiple agents on the same machine, use the `agent servicename` option with the installer to set a unique service name each time. For more information, see `installer_agent servicename`.

**Note**

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

**Note**

There is no 64-bit agent installation for Microsoft Windows platform.

To install the Agent, do the following:

1. Run the installer as required by your operating system.
2. The **Language Selection** dialog is displayed. Select your language, and click **OK**.

The Installation directory dialog is displayed.

3. Either change the installation directory, or accept the default value, and choose the connection type for the agent.
 - **Installation directory:** enables you to change the installation path.
 - **TCP/IP:** select if the agent uses TCP/IP to connect to the monitored database. This option is not available on Microsoft Windows platforms. TCP/IP is used by default.
 - **Socket:** select if the agent uses socket to connect to the monitored database. This is only possible if the agent is monitoring a local database. This option is not available on Microsoft Windows platforms.

If you choose Socket, you must enter the path to the socket later in the installation process.

Click **Forward**. The **Monitoring Options** dialog is displayed. The installation starts and the files are copied to the installation directory.

4. You can choose whether to monitor the host on which the agent is installed, or the host and a MySQL instance. If you select host only, you have to configure the connection to the MySQL Enterprise Service Manager, but no other configuration is required. If you select host and database, you must also configure the database connection parameters.

Click **Forward**. If you are installing on Apple OS X, the **Install as a service** dialog is displayed. This dialog enables you to install the agent as a service, which restarts each time the host is restarted. This option requires an Administrator's password.

On all other platforms, the **MySQL Enterprise Monitor Options** dialog is displayed.

5. The **MySQL Enterprise Monitor Options** dialog is displayed. Complete the following:
- **Hostname or IP address:** the hostname or IP address of the server where the MySQL Enterprise Service Manager is installed.
 - **Tomcat SSL Port:** the SSL port the MySQL Enterprise Service Manager is listening on.
 - **Agent Username:** the agent username. This is the username all agents must use to connect to the MySQL Enterprise Service Manager.
 - **Agent Password:** the agent's password. This is the password all agents must use to connect to the MySQL Enterprise Service Manager.
 - **Re-enter:** re-enter the agent's password.

Click **Forward**. The **Monitored Database Configuration Options** dialog is displayed.

6. The **Monitored Database Configuration Options** enables you to choose the remaining steps of the installation. The following options are available:
- **Validate hostname, port, and Admin account privileges:** select this option to attempt a test connection to the database with the supplied credentials, defined in the **Monitored Database Information** dialog. If you do not select this option, the installer does not attempt a test connection to the database with the supplied credentials.

**Note**

It is recommended to validate the connection.

- **Configure encryption settings for user accounts:** select this to configure the **Encryption Settings** dialog. If selected, this dialog is displayed after the **Monitored Database Information** dialog. The **Encryption Settings** dialog enables you to define the SSL options for connections to SSL-enabled MySQL Instances.

**Important**

Ensure the MySQL instance is SSL-enabled.

- **Configure less-privileged user accounts:** select if you intend to define the less-privileged user accounts, [General](#) and [Limited](#).

Make your selection and click **Forward**.

The **Monitored Database Information** dialog is displayed.

7. The **Monitored Database Information** dialog enables you to define the connection parameters for the MySQL instance the agent will monitor.

- **MySQL hostname or IP address:** the IP address or hostname of the server on which the MySQL instance is running.
- **MySQL Port:** the port the MySQL instance is listening on.
- **Admin User:** the admin user the agent uses. This can be the `root` user, or another user with the `SUPER` privilege.
- **Admin Password:** the password of the admin user.
- **Re-enter Password:** re-enter the admin user's password.
- **Monitor Group:** the group to which you want the instance to be added in MySQL Enterprise Service Manager. If the group does not exist, it is created, and the monitored instance added to it.

Click **Forward**. If you selected **Validate hostname, port, and Admin account privileges** on the **Monitored Database Configuration Options** dialog, the supplied credentials are verified against the MySQL instance.

If you selected **Configure encryption settings for user accounts** on the **Monitored Database Configuration Options** dialog, the **Encryption Settings** dialog is displayed.

8. The **Encryption Settings** dialog enables you to define the SSL connection parameters for your connection to the SSL-enabled MySQL Instance.
 - **Require Encryption:** enforces encrypted connections between the agent and the MySQL instance.
 - **Allow Self-Signed Certificates:** specifies whether self-signed certificates are permitted.
 - **CA Certificate:** the path to the CA certificate.
9. If you selected **Configure less-privileged user accounts** on the **Monitored Database Configuration Options**, the **Less Privileged User Account Creation** dialog is displayed.
 - **Auto-create Less Privileged Users:** select to automatically create the users, using the credentials supplied. If you do not create these users, all agent queries are run as the Admin user.
 - **General Username:** username of the General user.
 - **General Password:** password of the General user.
 - **Limited Username:** username of the Limited user.
 - **Limited Password:** password of the Limited user.

Click **Forward** to create the Agent account and complete the installation.

5.5 Starting/Stopping the MySQL Enterprise Monitor Agent

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

5.5.1 Starting/Stopping the Agent on Windows

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

**Note**

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

**Warning**

To report its findings, the agent needs to be able to connect to the 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 5.10, “Troubleshooting the Agent”](#).

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

```
shell> sc start MySQLEnterpriseMonitorAgent
```

or:

```
shell> net start MySQLEnterpriseMonitorAgent
```

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

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

You 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.

5.5.2 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 5.10, "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 5.10, “Troubleshooting the Agent”](#).

5.5.3 Starting/Stopping the Agent on Unix

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

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

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

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

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

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

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

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

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 5.10, “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 5.10, “Troubleshooting the Agent”](#).

5.5.4 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.

5.6 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 [Chapter 15, MySQL Instances Dashboard](#).

5.7 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 [Chapter 17, Replication Dashboard](#).

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 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](#).

5.8 Monitoring Outside the Firewall with an SSH Tunnel

If you run an SSH server on the machine that hosts the MySQL Enterprise Service Manager and an SSH client on the machine that hosts the agent, you can create an SSH tunnel so that the agent can bypass your firewall. First, you need to make an adjustment to the [agent-mgmt-hostname](#) value specified in the [etc/bootstrap.properties](#) configuration file. 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 5.5.1, “Starting/Stopping the Agent on Windows”](#).
- Under Unix see, [Section 5.5.3, “Starting/Stopping the Agent on Unix”](#).
- Under Mac OS X see, [Section 5.5.2, “Starting/Stopping the Agent on Mac OS X”](#).

5.9 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.

5.10 Troubleshooting the Agent

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

- Windows: [Section 5.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix: [Section 5.5.3, “Starting/Stopping the Agent on Unix”](#)
- Mac OS X: [Section 5.5.2, “Starting/Stopping the Agent on Mac OS X”](#)

Some additional tips are:

- 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.
- An easy way to confirm that the agent can log in to the service manager is to type **https://Monitor_UI_Host:18443/heartbeat** into the address bar of your web browser, substituting the appropriate host name and port. When the HTTP authentication dialog box opens, enter the agent user name and password. The following message indicates a successful login:

```
<exceptions>
<error>E0401: NullAgentPayloadException: []</error>
```

</exceptions>



Note

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

If you can log in successfully in the way described above and the agent is running, then there may be errors in Agent's configuration. Compare the host name, port, agent name, and password used in the MySQL Enterprise Monitor User Interface, and also check it using `agent.sh`, with the values you entered into the address bar of your web browser.

- If HTTP authentication fails then you are using incorrect credentials for the agent. Attempting to log in to the service manager using incorrect credentials creates an entry in the agent log file.

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 5.8, “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 5.7, “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.

5.11 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 10MB of monitored data in active RAM.

- Monitoring one MySQL instance: the agent backlog can store up to 40 minutes of monitored data before the backlog cache is filled and data dropped.
- Monitoring 10 MySQL instances: the agent backlog can store up to 4 minutes of monitored data before the backlog cache is filled and data dropped.

Chapter 6 Upgrading MySQL Enterprise Monitor Installations

Table of Contents

6.1 General considerations when upgrading MySQL Enterprise Monitor	49
6.2 Upgrading to MySQL Enterprise Monitor 3.1.x	49
6.3 Restoring from Backup	51

This chapter describes the upgrade process from MySQL Enterprise Monitor 3.0.x, or 2.3.x, to MySQL Enterprise Monitor 3.1.x.

The installation process is identical for both types of upgrade.

6.1 General considerations when upgrading MySQL Enterprise Monitor

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.

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.



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.

6.2 Upgrading to MySQL Enterprise Monitor 3.1.x

To upgrade from 2.3.x or 3.0.x to 3.1, you must upgrade your MySQL Enterprise Service Manager installation first, and your agents after the MySQL Enterprise Service Manager installation completes successfully.



Important

Due to changes in the TLS support introduced in MySQL Enterprise Monitor 3.0.22, you must upgrade your 3.0 installation (Service Manager and Agents) to at least version 3.0.22 before upgrading to 3.1.0. If you attempt to upgrade your

MySQL Enterprise Service Manager from a version earlier than 3.0.22 to 3.1, the MySQL Enterprise Service Manager will be unable to communicate with the MySQL Enterprise Monitor Agents due to the mismatch in supported SSL ciphers.

Upgrade Installer

The name of the upgrade file varies, but includes the target operating system, the version installed by the upgrade, and the component name. For example, a file named *mysqlenterprisemanager-3.1.1-windows-update-installer.exe* updates MySQL Enterprise Service Manager on Windows to version 3.1.1.

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.

For more information on the unattended upgrade process, see [Section 8.1.2, “MySQL Enterprise Service Manager Options”](#).

Service Manager Upgrade Process

1. Select the required installation language and click **Forward**. The **Installation Directory** dialog is displayed.
2. Confirm the location of your existing installation and click **Forward**. The **Backup of Previous Installation** dialog is displayed.



Note

If you are upgrading from 2.3, an alert is displayed regarding data which cannot be migrated from 2.3 to 3.1.

3. If you want to backup your existing installation, select Yes and edit the **Backup directory** field if required.

If you do not want to backup your existing installation, select **No**.

Click **Forward** to continue.

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

4. Confirm the values in the **Tomcat Server Port** and **Tomcat SSL Port** fields.

Click **Forward** to continue.

The **Repository Configuration** dialog is displayed.

5. Confirm your repository Configuration. Click **Forward** to continue.

The upgrade is now ready to install. Click **Forward** to upgrade your installation, or **Back** to review or edit any values.

The upgrade process shuts down the MySQL Enterprise Service Manager services and performs the backup of the existing installation, if you chose to do so, then copies the new files to the installation directory, and starts the new services.

6. As the installation completes, you are prompted to reimport any SSL certificates which may have existed in the previous installation. The installer also provides information on where the certificates were backed up.

Click **OK** to continue.

7. The installation completes. You are prompted to launch and configure the application.

Agent Upgrade to 3.1

1. Select the required installation language and click **Forward**. The **Installation Directory** dialog is displayed.
2. Confirm the location of your existing installation.
3. Confirm the location of your existing installation and click **Forward**. The **Backup and Restart Options** dialog is displayed.

- **Backup the current installation** checkbox, specify an alternate location if required. This option is enabled by default.

If you do not want to back up your existing installation, deselect this checkbox.

- **Restart Agent immediately after updating all files** checkbox. Enabled by default.

If you want to start your agent manually, at a later time, deselect this checkbox.

The installation is started and completes.

6.3 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.

Chapter 7 Post-installation Considerations

Table of Contents

7.1 General Considerations	53
7.2 Installing SSL Certificates	54
7.3 Changing an SSH Host Key	57

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

7.1 General Considerations

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

New Users

1. **Groups and Connections:** Groups have always been used to define Event handling and Advisor scheduling policies; in this release Groups can also be used to restrict visibility and access to specific MySQL instances and their hosts. Before you create Connections and set up Groups, we recommend you first read the note immediately following on Users, Roles, and Access Control
 - To create a Connection, select **MySQL Instances** from the **Dashboard** menu. Create new monitoring connections either by processing the unmonitored instances already discovered by MEM or by manually specifying connection parameters for each MySQL Instance you will monitor. See [Section 15.2, “MySQL Instance Details”](#) for more information on creating connections in the User Interface.
 - Use the new Group Editor on the Settings menu to collect your MySQL instances into Groups.
2. **Users, Roles, and Access Control (ACLs):** Before using the Settings menu to create User accounts, see [Chapter 23, Access Control](#) and [Chapter 24, Access Control - Best Practices](#).

Will you provide open access to all monitored resources for all Users? Or will you define Roles granting access to specific groups of MySQL Instances? If you intend to restrict access in this way, you must first create Groups of MySQL instances, see [Chapter 16, Managing Groups of Instances](#). Only after you create groups can you create group-specific Roles.

Finally, assign users to your Roles.

You can also map users to Roles defined in LDAP or Active Directory.

3. **Configure Event Handling and Notification policies:** Open **Event Handling** from the Settings menu. Complete, and test, the SMTP, or SNMP, configuration. See [Chapter 20, Events and Event Handling](#) for more information.
4. **Overview Dashboard:** Select **Overview** from the **Dashboard** menu. Set the defaults for the groups you want to view, the time range, and graphs to display.
5. **Advisors:** You can accept the defaults defined, or select **Advisors** from the **Settings** menu and customize the threshold for groups, or individual MySQL Instances. For more information, see [Chapter 19, Advisors](#).

6. **SQL Performance Tuning** - If you are monitoring instances of MySQL running version 5.6.14 or later, make sure to see the rich SQL performance tuning data available in the Query Analyzer. (If you are monitoring earlier MySQL versions, make sure to download a Query Analyzer plugin so you can see SQL performance data as well.)
7. **I/O and Lock Contention** - If you are using MySQL 5.6 or later consider deploying the sys schema, and making use of the new Database File I/O and Lock Waits reports from the Reports & Graphs menu. These will help you identify who or what is using the most I/O, and whether there is any lock wait contention within your MySQL Instance. See the Database File I/O and Lock Waits reports documentation for more information.

Existing users: Guide to completing your upgrade

- **Update Agents:** If you have not done so already, we recommend updating your Agents before continuing. See [Chapter 6, Upgrading MySQL Enterprise Monitor Installations](#).
- **Users, Roles, and Access Control (ACLs):** This release introduces Access Control Lists (ACLs). Built-in Roles have replaced the privileges previously defined in Manage Users in version 3.0. Your system already has been migrated, however we strongly recommend reviewing [Chapter 23, Access Control](#) and [Chapter 24, Access Control - Best Practices](#).

You can continue to permit 3.0-style open access to all monitored resources for all Users who login; but you can now also define Roles that allow visibility and access to specific Groups of MySQL Instances and grant those Roles only to selected Users. If you're using external services like LDAP or Active Directory, you can optionally map users to roles you've defined there.

If you're going to use ACLs to restrict visibility or access, we suggest you review your existing Groups with that in mind. The new Group Editor is located in the Settings menu. For an explanation of how 3.0 privileges have been migrated to 3.1, see [Section 23.6, "Default Users and Roles"](#).

- **Overview Dashboard:** If you haven't already done so, select the Dashboards menu, click on Overview and set defaults for which Group you want to view, the graph time range, and the set and order of Graphs to display. See [Chapter 16, Managing Groups of Instances](#).
- **SQL Performance Tuning:** If you are monitoring instances of MySQL running version 5.6.14 or later, make sure to see the rich SQL performance tuning data available in the Query Analyzer. (If you are monitoring earlier MySQL versions, make sure to download a Query Analyzer plugin so you can see SQL performance data as well.)
- **I/O and Lock Contention:** If you are using MySQL 5.6 or later consider deploying the sys schema, and making use of the new Database File I/O and Lock Waits reports from the Reports & Graphs menu. These will help you identify who or what is using the most I/O, and whether there is any lock wait contention within your MySQL Instance. See [Section 18.2, "Database File I/O and Lock Waits"](#) for more information.

7.2 Installing SSL Certificates



Important

The self-signed certificates are generated by the MySQL Enterprise Monitor installation or upgrade process, and are set to expire after 365 days. 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.

Checking the Keystore

All certificates and keys are stored in the Tomcat keystore. To check the certificates stored in the keystore, run the following command:

```
keytool -keystore $INSTALL_ROOT/apache-tomcat/conf/keystore -list -v
```

Generating Keystore, Key, and Certificate

For information on using `keytool`, see [Java Keytool](#).

To generate the certificate and add it to the default keystore, run the following command:

```
$INSTALL_ROOT/java/bin/keytool -genkey -keyalg RSA -sigalg SHA256withRSA  
-keystore $INSTALL_ROOT/apache-tomcat/conf/keystore -alias tomcat  
-validity 365 -keysize 2048
```

This generates a 2048-bit, RSA private key, and certificate. Further, this is the same command as used by the MySQL Enterprise Monitor installers.



Important

When prompted for the key password, if you enter a password, rather than accepting the default by pressing Enter, you must also add the new password in the Tomcat configuration file, [server.xml](#).

MySQL Enterprise Service Manager SSL Import

To install an SSL certificate for the MySQL Enterprise Service Manager you must use the Java `keytool` to import the certificate into the keystore.

To import your certificate, run the following command:

```
keytool -import -trustcacerts -alias mycertificate -file cert.pem -keystore myKeystore
```

If you want to import an existing certificate, which is password protected, you must convert it to a format understood by the Java keytool. The certificate must be converted from X509 to pkcs12 format using the openssl toolkit and the following command:

```
openssl pkcs12 -export -in [path-to-x509Cert] -inkey [path-to-cert-private-key]  
-out [path-to-cert-to-import-for-keystore] -name tomcat
```



Important

The certificate name must be set to Tomcat, or match the name used in the key generation steps.

To import the converted certificate, run the following command:

```
$INSTALL_ROOT/java/bin/keytool -importkeystore
-srckeystore [path-to-cert-to-import-for-keystore] -srcstoretype pkcs12
-destkeystore $INSTALL_ROOT/apache-tomcat/conf/keystore
-deststoretype jks -srcalias tomcat -destalias tomcat
```

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](#).

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 7.1 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-verify-host-certs</code> is <code>true</code>	
<code>ssl-ca-keystore-password</code>	String	Password for the CA keystore, if <code>ssl-verify-host-certs</code> is <code>true</code>	

An example `bootstrap.properties` SSL certification section:

```
ssl-verify-hostname=false
ssl-verify-host-certs=true
ssl-ca-keystore-path=/path/to/keystore
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/ca
```

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.

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
```

7.3 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 alerts 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 8 Unattended Installation Reference

Table of Contents

8.1 Unattended Installation	59
8.1.1 Performing an Unattended Installation	59
8.1.2 MySQL Enterprise Service Manager Options	60
8.1.3 MySQL Enterprise Monitor Agent Options	66

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 `installbuilder.sh` within the installation package directory. For example:

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

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

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

8.1.2 MySQL Enterprise Service Manager Options

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

Table 8.1 MySQL Enterprise Service Manager Installer Options

Format	Description
<code>--adminpassword</code>	Password for the database repository

Format	Description
<code>--adminuser</code>	Username for the database repository
<code>--backupdir</code>	Backup directory path.
<code>--createDataBackup</code>	Backup stored data. Upgrade process only.
<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
<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	--backupdir	
Permitted Values	Type	string

Upgrade only. The backup directory.

- `--createDataBackup`

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

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

- `--debuglevel`

Command-Line Format	<code>--debuglevel</code>	
Permitted Values	Type	<code>numeric</code>
	Default	<code>2</code>
	Min Value	<code>0</code>
	Max Value	<code>4</code>

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	<code>string</code>
	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	<code>string</code>
	Default	<code>/Applications/mysql/enterprise/monitor/</code>
Permitted Values (Unix)	Type	<code>string</code>
	Default	<code>/opt/mysql/enterprise/monitor/</code>
Permitted Values (Windows)	Type	<code>string</code>
	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	<code>string</code>
	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>63</code>

		medium (Up to 100 MySQL Servers monitored from a medium-sized, but shared, server with 4 to 8GB of RAM.)
		large (More than 100 MySQL Servers monitored from a high-end, dedicated server, with more than 8GB RAM.)

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

- [--tomcatport](#)

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

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

- [--tomcatsslport](#)

Command-Line Format	--tomcatsslport	
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	--mysql-identity-source	
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	<code>boolean</code>
	Default	<code>0</code>
	Valid Values	<code>0</code> (Do not use SSL when connecting to the database) <code>1</code> (Use SSL when connecting to the database)

Use SSL when connecting to the database.

- `--adminuser`

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

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 are stored in the `config.properties` configuration file.

- `--unattendedmodeui`

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

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	<code>string</code>
	Default	<code>bundled</code>

	Valid Values	bundled (Use the bundled MySQL server)
		existing (Use an existing (user supplied) MySQL server)

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

- [--dbport](#)

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

The TCP/IP port for the MySQL database used to store MySQL Enterprise Service Manager repository data. If you 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	--dbhost	
Permitted Values	Type	string
	Default	127.0.0.1

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

- [--dbname](#)

Command-Line Format	--dbname	
Permitted Values	Type	string
	Default	mem

The name of the MySQL Enterprise Service Manager repository.

- [--forceRestart](#)

Command-Line Format	--forceRestart	
Permitted Values	Type	boolean
	Default	0
	Valid Values	0 (Do not restart services)
		1 (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
--proxyservicename	Service name for the Proxy

- `--agentpassword`

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

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

- `--createBackup`

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

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	<code>boolean</code>
	Default	<code>1</code>

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	<code>string</code>
	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	<code>string</code>
	Default	<code>yes</code>

	Valid Values	yes (Check host)
		no (Do not check host)

Validate the MySQL hostname or IP address

- [--debuglevel](#)

Command-Line Format	--debuglevel	
Permitted Values	Type	numeric
	Default	2
	Min Value	0
	Max Value	4

Set the debug information level for log data written to the file specified by [debugtrace](#).

- [--debugtrace](#)

Command-Line Format	--debugtrace	
Permitted Values	Type	string

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

- [--installdir](#)

Command-Line Format	--installdir	
Permitted Values (OS X)	Type	string
	Default	/Applications/mysql/enterprise/agent/
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	--installer-language	
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	--managerhost
----------------------------	-------------------------------

Permitted Values	Type	<code>string</code>
-------------------------	-------------	---------------------

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

- `--managerport`

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

Tomcat SSL Port

- `--mode`

Command-Line Format	<code>--mode</code>	
Permitted Values (Linux)	Type	<code>string</code>
	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	<code>string</code>
	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	<code>string</code>
	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	<code>string</code>
	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	<code>string</code>
	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	<code>string</code>
	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	<code>string</code>

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	<code>numeric</code>
	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	<code>string</code>

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	<code>string</code>

An MySQL Server administrative user for the MySQL instance that will be monitored. This user must already exist.

- `--agent_autocreate`

Command-Line Format	<code>--agent_autocreate</code>	
Permitted Values	Type	<code>boolean</code>

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	<code>string</code>

The username for the general user.

- `--generalpassword`

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

Password for the `--generaluser`.

- `--limiteduser`

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

The username for the limited user.

- `--limitedpassword`

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

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	<code>string</code>
	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.

- `--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	<code>string</code>
	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`

Command-Line Format	<code>--ignore-old-proxy-aggr</code>	
Permitted Values	Type	<code>boolean</code>
	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	<code>string</code>

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	mysql-monitor-agent
Permitted Values (OS X)	Type	string
	Default	mysql.monitor.agent
Permitted Values (Unix)	Type	string
	Default	mysql-monitor-agent
Permitted Values (Windows)	Type	string
	Default	MySQL Enterprise Monitor Agent

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	75
9.2 Tuning CPU	77
9.3 Tuning Apache Tomcat Threads	78
9.4 Tuning Agent Memory Requirements	79

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	C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\bin\setenv.bat
Linux / Solaris	/opt/mysql/enterprise/monitor/apache-tomcat/bin/setenv.sh
Mac OS X	/Applications/mysql/enterprise/monitor/apache-tomcat/bin/setenv.sh

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.

**Important**

`MaxPermSize` is not supported in Java 8. This parameter is not present in new installations of MySQL Enterprise Service Manager, but is not removed by the upgrade process. As a result, a log message is generated explaining the deprecation of the parameter.

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 --Jvms 768 ++JvmOptions="-XX:+UseParallelOldGC"
++JvmOptions="-XX:+HeapDumpOnOutOfMemoryError"
++JvmOptions="-XX:HeapDumpPath=@@BITROCK_TOMCAT_ROOTDIR@@\temp"
++JvmOptions="-XX:MaxPermSize=512M"
```

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

Tuning InnoDB Memory

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

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

Table 9.3 MEM repository configuration file location (default)

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

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 allows 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	<code>C:\Program Files\MySQL\Enterprise\Monitor\bin\config.bat</code>
Linux / Solaris	<code>/opt/mysql/enterprise/monitor/bin/config.sh</code>
Mac OS X	<code>/Applications/mysql/enterprise/monitor/bin/config.sh</code>

This Service Manager repository tool has the following options:

Option	Description
-----	-----
<code>--accept-keystore-password, --akp</code>	If specified, the user will be asked the keystore password, otherwise the password "changeit" will be used to access the keystore
<code>--help</code>	Prints this usage message
<code>--md, --mysql-db</code>	MySQL database for the Service Manager repository
<code>--mp, --mysql-port</code>	MySQL port for the Service Manager repository
<code>--ms, --mysql-server</code>	MySQL server for the Service Manager repository
<code>--mu, --mysql-user</code>	MySQL username for the Service Manager repository
<code>--new-install, --ni</code>	Generates a keystore with a fresh self-signed certificate for a new installation
<code>--tbp, --tomcat-backup-path</code>	Tomcat backup path to be used to perform upgrade
<code>--upg, --upgrade</code>	Upgrades certificates on an existing non OS X installation
<code>--upgo, --upgrade-osx</code>	Upgrades keystore on an existing OS X
<code>-v, --version</code>	Displays the version of the agent and components

Updating MySQL Enterprise Service Manager Configuration

Option	Description
-----	-----
<code>--md, --mysql-db</code>	MySQL database for the Service Manager repository
<code>--mp, --mysql-port</code>	MySQL port for the Service Manager repository
<code>--ms, --mysql-server</code>	MySQL server for the Service Manager repository

```
--mu, --mysql-user MySQL username for the Service Manager repository
```

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.

Updating Security Configuration

Option	Description
-----	-----
--accept-keystore-password, --akp	If specified, the user will be asked the keystore password, otherwise the password "changeit" will be used to access the keystore
--new-install, --ni	Generates a keystore with a fresh self-signed certificate for a new installation
--tbp, --tomcat-backup-path	Tomcat backup path to be used to perform upgrade
--upg, --upgrade	Upgrades certificates on an existing non OS X installation
--upgo, --upgrade-osx	Upgrades keystore on an existing OS X

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:

```
<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 Configuration Utilities

Table of Contents

10.1 Service Manager Configuration Utilities	81
10.2 Agent Configuration Utility	83

This chapter describes the utilities delivered with MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent.

10.1 Service Manager Configuration Utilities



Note

The parameters listed here, with the exception of the four repository connection parameters, correspond to those displayed on the **Welcome to MySQL Enterprise Monitor** page used for initial setup. For more information, see [Section 13.1, “Initial Log-In”](#).

These parameters enable you to configure MySQL Enterprise Service Manager from script or command line.

The `config.sh` / `config.bat` script is used to configure the MySQL Server Repository for the Service Monitor. Its default location:

Table 10.1 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 the options.

The Service Manager config utility contains the following sets of commands:

- Service Manager Configuration Utilities: define or change the configuration of the MySQL Enterprise Service Manager.
- Certificate Utilities: modify or upgrade the MySQL Enterprise Service Manager SSL certificates.

Service Manager Configuration Utilities

The `config` script enables you to define or change any of the system configuration parameters such as credentials used to connect to the repository, proxy connection details, and MySQL Enterprise Service Manager user credentials.

Table 10.2 Service Manager Config Utilities

Name	Description
<code>--mysql-user=<value></code>	MySQL username for the Service Manager repository. The password is requested via STDIN

Name	Description
--mu=<value>	when the command is run. The default value is <code>service_manager</code> .
--mysql-port=<value>	MySQL port for the Service Manager repository. The port the target MySQL server listens on. The default is 13306.
--mp=<value>	
--mysql-db=<value>	MySQL database for the Service Manager repository. The name of the database used for the repository. The default is <code>mem</code> .
--md=<value>	
--mysql-server=<value>	MySQL server for the Service Manager repository. This must be a resolvable name or IP address of the server where the MySQL instance is running.
--ms=<value>	
--sm-admin-user=<value>	Service Manager manager username. The user defined here is added to the manager role.
--sm-agent-user=<value>	Service Manager agent username. The user defined here is added to the agent role.
--auto-update	Enable automatic checking for online updates.
--purge-quant=<value>	Defines the Query Analyzer data retention policy. Query Analyzer data older than the number of days defined here is deleted. Default is 28 days.
--purge-data=<value>	Defines the historical data retention policy. Historical data older than the number of days defined here is deleted. Default is 28 days.
--proxy-host=<value>	HTTP Proxy host
--proxy-port=<value>	HTTP Proxy port
--proxy-user=<value>	HTTP Proxy username



Important

Passwords are always requested via STDIN and are requested in the order manager, agent, and proxy, regardless of the order in which they are defined on the command line or in script.

The following example instructs the MySQL Enterprise Service Manager to use a locally installed instance, listening on port 3306, the `mem` database, and connect using the user `service_manager`:

```
config.sh --mysql-server=localhost --mysql-port=3306 --mysql-db=mem
--mysql-user=service_manager
```

The following is an example of a basic setup, defining the admin and agent users, only. :

```
config.sh --sm-admin-user=admin --sm-agent-user=agent
```

All other parameters are set to their default values.



Important

You are prompted to define passwords for each of the users defined. Passwords are only accepted through STDIN.

Passwords are always requested in the order manager, agent, proxy, regardless of the order defined on the command line or in the script.

All other values are set to their defaults.

The following is an example of a complete setup, defining all available options:

```
config.sh --sm-admin-user=admin --sm-agent-user=agent --purge-quan=7
          --purge-data=14 --proxy-host=localhost --proxy-port=9190
          --proxy-user=proxy --auto-update
```

Service Manager Certificate Utilities

This section describes the SSL certificate utilities.

Table 10.3 Service Manager Certificate Utilities

Name	Description
<code>--tomcat-backup-path=<value></code>	Tomcat backup path to be used to perform upgrade
<code>--tbp=<value></code>	
<code>--upgrade</code>	Upgrades certificates on an existing non OS X installation
<code>--upg</code>	Upgrades certificates on an existing non OS X installation
<code>--upgrade-osx</code>	
<code>--upgo</code>	
<code>--new-install</code>	Generates a keystore with a fresh self-signed certificate for a new installation
<code>--ni</code>	
<code>--accept-keystore-password</code>	
<code>--akp</code>	If specified, the user will be asked the keystore password, otherwise the password "changeit" will be used to access the keystore

10.2 Agent Configuration Utility

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

Table 10.4 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.

Agent Connection Utilities

This section describes the utilities available for agent connections and testing.

Table 10.5 Agent Connection Utility

Name	Description
<code>--test-credentials</code> <code>-T</code>	Test MySQL connection credentials.
<code>--test-privileges</code>	Test admin user's privileges to manage other users.
<code>--create-connection</code> <code>-c</code>	Create or Modify a MySQL connection.
<code>--delete-connection</code> <code>-d</code>	Close and Delete a MySQL connection (must also specify --connection-id)
<code>--show</code> <code>-s</code>	Show information about all MySQL connections on this agent
<code>--auto-manage-extra-users</code> <code>-m</code>	Auto-create general / limited users (Actions: Create, Modify)
<code>--host=<value></code> <code>-h <value></code>	Host for the MySQL instance (Actions: Create, Modify)
<code>--port=<value></code> <code>-P <value></code>	Port for the MySQL instance (Actions: Create, Modify)
<code>--socket=<value></code> <code>-S <value></code>	Socket for the MySQL instance (Actions: Create, Modify)
<code>--limited-user=<value></code> <code>-l <value></code>	Limited level credentials (Actions: Create, Modify)
<code>--general-user=<value></code> <code>-k <value></code>	General user credentials
<code>--admin-user=<value></code> <code>-j <value></code>	Admin user credentials
<code>--connection-id=<value></code> <code>-i <value></code>	Connection ID
<code>--connection-group=<value></code> <code>-g <value></code>	MEM Group to use for created/modified connection
<code>--force-plain-stdin,</code> <code>-f</code>	Force the use of STDIN for password inputs (password input is not masked - this option is useful only for very specific uses of these utilities, like calls from within automated scripts)
<code>--disable-topology-discovery</code>	Disable replication topology discovery. Use this parameter if you are not using replication, or if you want to discover the topology at a later time. Topology discovery can be time-consuming.

Name	Description
<code>--mysql-identity-source=<value></code>	Source of identity for the MySQL instance for this connection, <code>default</code> or <code>host_plus_datadir</code> . <code>default</code> uses either the <code>server_uuid</code> variable, if present, or generates a new uuid. <code>host_and_datadir</code> uses a hash of the host identity and the path to the MySQL instance's data directory to create a unique identity.
<code>--require-encryption</code>	Require the use of TLS for the MySQL connection.
<code>--allow-self-signed-certs</code>	When using <code>--require-encryption</code> , allow self-signed TLS certificates.
<code>--ca-file-path=<value></code>	When using <code>--require-encryption</code> , but using a private certificate authority, the path to the CA file.

The following example tests credentials for the root user on localhost:3306:

```
agent.bat --test-credentials --admin-user=root --host=localhost --port=3306
```

The following example creates a connection using only the admin user for localhost:3306:

```
agent.bat -c --admin-user=root --host=localhost --port=3306
```

The following example creates a connection, using only the admin user, to localhost:3306, and forces STDIN password:

```
agent.bat -c --admin-user=root --host=localhost --port=3306 -f
```

The following example creates a connection, using only the admin user, to localhost:3306, and add to the groups Standard, Special, and Third:

```
agent.bat -c --admin-user=root --host=localhost --port=3306
--connection-group=Standard --connection-group="Special Group"
--connection-group="Third Group"
```

Agent Configuration Utilities

This section describes the utilities available for agent configuration.

Table 10.6 Agent Configuration Utility

Name	Description
<code>--agent-user=<value></code>	Set the credentials that the Agent uses to connect to the Service Manager
<code>-u <value></code>	
<code>--url=<value></code>	Set the URL for the Service Manager
<code>-U <value></code>	
<code>--uuid=<value></code>	Set the Agent UUID
<code>-I <value></code>	
<code>--agent-group=<value></code>	Set the MEM Group to use for all MySQL connections from this Agent

Name	Description
<code>-G <value></code>	
<code>--force-plain-stdin</code> <code>-f</code>	Force the use of STDIN password inputs (password input is not masked - this option is useful only for very specific uses of these utilities, such as calls from automated scripts)
<code>--run-collection-tests</code> <code>-t</code>	Discover, and attempt to collect OS related assets and dump them to STDOUT (for debugging)

The following example sets the user name and URL used by the agent to connect to the MySQL Enterprise Service Manager:

```
agent.sh --agent-user=agent --url=https://localhost:8443
```

Chapter 11 Proxy and Aggregator Installation

Table of Contents

11.1 Proxy Aggregator Architecture	87
11.2 Prerequisites	88
11.3 Installing the Proxy and Aggregator	89
11.4 Graphical Installation Wizard	89
11.5 Text-Based Installation	91
11.6 Unattended Installation	92
11.7 Starting and Stopping the Proxy and Aggregator	95
11.8 Configuration Options	96

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:

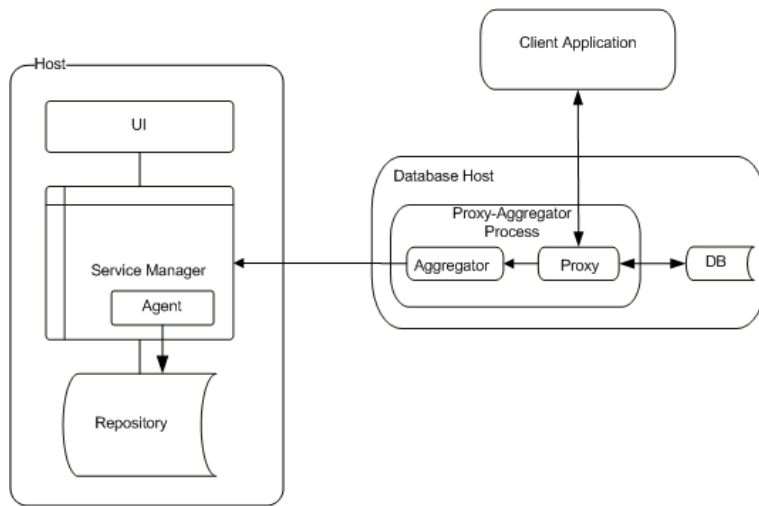
- **MySQL Enterprise Monitor Proxy:** the Proxy functions as the communications chassis for the Aggregator and is responsible for intercepting the communications between the client application and the MySQL instance. This enables the Aggregator to collect the raw query data sent from the client application 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



Important

No installer is provided for MySQL Enterprise Monitor Proxy and Aggregator on Windows platforms at this time.

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: • <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

Option	Description
	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: • Windows: <code>Win32</code> • OS X: <code>osx</code> • Solaris: <code>xwindow</code> • Linux: <code>gtk</code> (Default) and <code>xwindow</code>. <code>--mode</code> can also initiate text mode and unattended installations. • <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>	Sets the graphical elements to use, if any, in the unattended installation. The following options are available: • <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>	Specifies which component to install. The following options are available: • <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>	Sets the connection method used by the proxy to connect to the monitored MySQL instance. The following options are available:

Option	Description
	<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>--aggregator servicename</code>	Sets the unique name for the Aggregator service. Default value is <code>mysql-monitor-aggregator</code> .
<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.

Option Name	Description
<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 LUA_PATH.
<code>--lua-cpath=<path></code>	Sets the LUA_CPATH

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

Option Name	Description
	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>	The host name (or IP address) and port of the MySQL server to connect to. You can specify multiple backend servers by supplying multiple options. Clients are connected to each backend server in round-robin fashion. For example, if you specify two servers A and B, the first client connection will go to server A; the second client connection to server B and the third client connection to server A. When using this option on the command line, you can specify the option and the server multiple times to specify multiple backends. When using this option in a configuration file, separate multiple servers with commas.
<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 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 proxy-backend-addresses 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	101
12.2 Using the MySQL Enterprise Plugin for Connector/J	105
12.3 Using the MySQL Enterprise Plugin for Connector/Net	110

This section describes how to configure the Connectors to pass query information to the Query Analyzer. The following configurations are described:

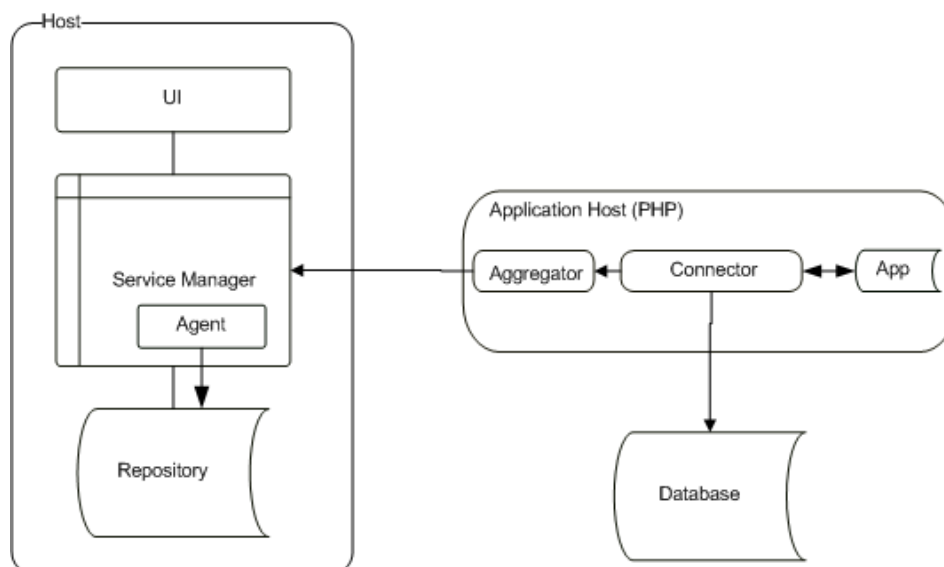
- [Section 12.1, “Using the MySQL Enterprise Plugin for Connector/PHP”](#): describes the configuration of a PHP-based application which uses the MySQL Enterprise Plugin for Connector/PHP and MySQL Enterprise Monitor Aggregator to feed query information to the Query Analyzer
- [Section 12.2, “Using the 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>	Timeout, in seconds, for communications with the MySQL Enterprise Monitor Aggregator. - Property type: <code>integer</code> - Default value: <code>1</code> This property can be combined with the <code>aggregator_connect_timeout_usec</code> property.
<code>aggregator_connect_timeout_usec</code>	Timeout, in microseconds, for communications with the MySQL Enterprise Monitor Aggregator. - Property type: <code>integer</code> - Default value: <code>0</code> This property can be combined with the <code>aggregator_connect_timeout_sec</code> property.
<code>aggregator_user</code>	The Aggregator's username. See Chapter 11, Proxy and Aggregator Installation for more information. Property type: <code>string</code>
<code>aggregator_password</code>	The Aggregator's password. Property type: <code>string</code>
<code>aggregator_url</code>	The IP address, or hostname, and port of the Aggregator installation. - Property type: <code>string</code> - Default value: <code>tcp://127.0.0.1:14000</code>
<code>debug_callback</code>	This property should be used only when debugging your MySQL Enterprise Monitor installation with MySQL Support personnel. Defines the name of the callback function to invoke when data is sent to the Aggregator. The callback is defined in the PHP

Property	Description
	application and is a function which requires a single parameter, the array of HTTP requests made to the Aggregator.
<code>disable_backtrace</code>	Defines whether a backtrace is performed. Backtrace is useful for debugging but has a performance impact. - 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>	Defines the location of a log file which logs all query information sent to the Aggregator. This should only be used for debugging purposes because every request is logged, resulting in a very large log file. Property type: <code>string</code>
<code>quan_enabled</code>	Defines whether query analysis is enabled. - 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

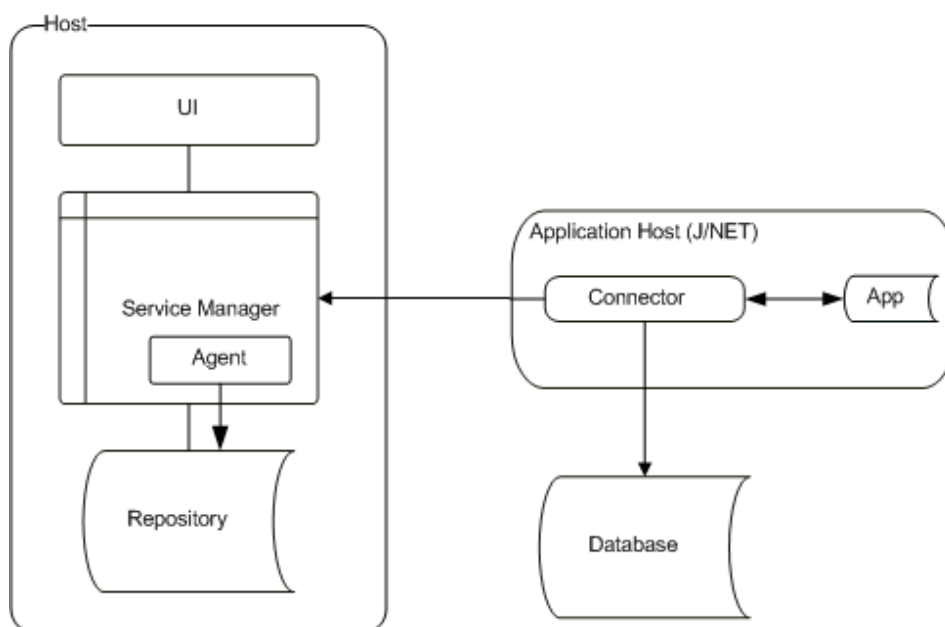
The MySQL Enterprise Plugin for Connector/J, enables query analysis for your applications without requiring any modification to the application code.



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.StatementPerformanceCou</code>
<code>disableSourceLocation</code>	Defines whether to send stack traces with example queries to MySQL Enterprise Service Manager. - Property type: <code>boolean</code> - Values: <code>true</code> or <code>false</code> (default).
<code>serviceManagerUrl</code>	Defines the URL of the MySQL Enterprise Service Manager. Include the full URL and port number. - Property type: <code>string</code> - Value: URL and port number of MySQL Enterprise Service Manager.
<code>serviceManagerUser</code>	Defines the Agent username to use when connecting to MySQL Enterprise Service Manager. Property type: <code>string</code>
<code>serviceManagerPassword</code>	Defines the Agent password to use when connecting to MySQL Enterprise Service Manager. Property type: <code>string</code>
<code>serviceManagerConnectTimeout</code>	Defines the number of seconds to wait for a connection to MySQL Enterprise Service Manager. - Property type: <code>numeric</code> - Default value: 0
<code>serviceManagerResponseTimeout</code>	Defines the number of seconds to wait for a response from MySQL Enterprise Service Manager.

Property Name	Description
	- Property type: numeric - Default value: 0
mysqlServerUUID	If you are unable to retrieve the server's UUID, define it with this property. Property type: string To retrieve the UUID, the plugin requires SELECT privileges on mysql.inventory.

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
verifySslHostnames	If set to true, it enables verification of the host names in the SSL Server certificate. Host names are verified using the same schema as used by Firefox, and Curl, and specified by RFC 2818. - Property type: boolean - Default value: false
verifySslCerts	Defines whether the plugin verifies the certificate presented by the server was signed by a CA in the trustCertificateKeystore. - Property type: boolean - Default: false, verification is disabled.
trustCertificateKeystoreUrl	Defines the URL of the trusted root certificate KeyStore. If none is specified, the Java defaults are used. Property type: string
trustCertificateKeystorePassword	Defines the password for the KeyStore. Property type: string
trustCertificateKeystoreType=[type]	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.
clientCertificateKeystoreUrl	Defines the URL of the client KeyStore. If none specified, Java defaults are used.

Property Name	Description
<code>clientCertificateKeystorePassword=[password]</code>	Defines the password to use for the client certificate store.
<code>clientCertificateKeystoreType</code>	Defines the KeyStore type for client certificates. If type is set to NULL or empty, JKS is used by default.

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.etoools.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)

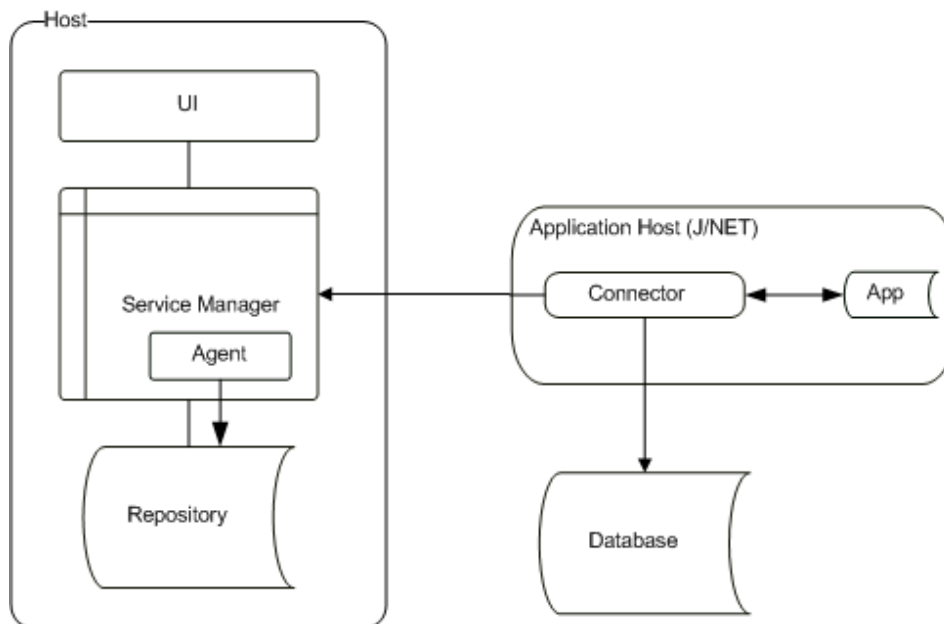
Further configuration of the plugin is done via the Enterprise Service Manager's user interface, using the **Query Analyzer** tab. From here, the capture of query performance data for a given MySQL instance that the plugin is being with can be enabled or disabled. Further, the level of detail that is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

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 27.3, “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 27.3, "Query Analyzer User Interface"](#).

Part III Using MySQL Enterprise Monitor

Table of Contents

13	User Interface	117
13.1	Initial Log-In	117
13.2	Setting the Timezone and Locale	118
13.3	Menus and Toolbars	118
13.3.1	Main Menus	118
13.3.2	Status Summary	120
14	Overview	123
14.1	Database Statistics	123
14.2	Overview Graphs	124
14.3	General Database Statistics	125
14.4	Group Overview Configuration	125
15	MySQL Instances Dashboard	127
15.1	MySQL Instance Dashboard UI	127
15.2	MySQL Instance Details	129
15.3	Adding Instances	131
15.3.1	Adding a MySQL Instance	131
15.3.2	Adding Multiple MySQL Instances	135
15.4	Filtering MySQL Instances	135
16	Managing Groups of Instances	137
17	Replication Dashboard	139
18	Reports and Graphs	143
18.1	All Timeseries Graphs	143
18.1.1	Graph Controls	143
18.1.2	Graph Types	145
18.2	Database File I/O and Lock Waits	145
18.2.1	sys Schema	145
18.2.2	Database File I/O Graphs and Reports	146
18.2.3	Lock Waits Report	148
18.3	InnoDB Buffer Pool Usage	148
19	Advisors	151
19.1	Advisors Page	151
19.2	Advisor Types	155
19.3	Advisor Thresholds	156
19.4	Advisor Schedules	158
20	Events and Event Handling	159
20.1	Events	159
20.2	Event Handling	163
20.2.1	Event Handlers	163
20.2.2	Event Handling Page	163
20.3	Creating Event Handlers	168
20.3.1	Event Action Log	170
20.3.2	Suspending an Event Handler	170
21	Expression-Based Advisor Reference	171
21.1	Administration Advisors	171
21.2	Agent Advisors	177
21.3	Availability Advisors	178
21.4	Cluster Advisors	179
21.5	Memory Usage Advisors	181
21.6	Monitoring and Support Services Advisors	183
21.7	Operating System Advisors	184
21.8	Performance Advisors	184

21.9 Replication Advisors	190
21.10 Schema Advisors	195
21.11 Security Advisors	199
22 GUI-Based Advisor Reference	207
22.1 Agent Health Advisor	207
22.2 MySQL Enterprise Backup Health Advisor	210
22.3 MySQL Process Discovery Advisor	211
22.4 Duplicate MySQL Server UUID	212
22.5 HTTP Server KeyStore's Certificate About to Expire	212
22.6 sys Schema Install Advisor	212
22.7 CPU Utilization Advisor	213
22.8 Filesystem Free Space Advisor	214
22.9 MySQL Process	216
22.10 Query Analysis Advisors	216
22.11 Security Advisors	217
23 Access Control	219
23.1 Users and Roles	219
23.2 Permissions	220
23.3 Monitored Assets Permissions	220
23.3.1 Server Group	221
23.3.2 MySQL Instances	221
23.4 Monitoring Services	223
23.5 MySQL Enterprise Monitor	223
23.6 Default Users and Roles	225
23.7 Creating Users and Roles	226
24 Access Control - Best Practices	229
24.1 Open Permission Sets	230
24.2 Strict Permission Set	232
25 Global Settings	237
25.1 Server Locale	237
25.2 Server Hostname	237
25.3 Customize MySQL Server Name	237
25.4 Data Purge Behavior	239
25.5 My Oracle Support Credentials	239
25.6 HTTP Proxy Settings	240
25.7 External Authentication	240
26 Customizing MySQL Enterprise Monitor	245
26.1 Creating Advisors and Rules	245
26.1.1 Creating Advisors	245
26.1.2 Overview of Graph Creation	246
26.1.3 Overview of Advisor Creation	247
26.1.4 Variables	248
26.1.5 Thresholds	248
26.1.6 Using Strings	249
26.1.7 Wiki Format	250
26.1.8 Creating a New Advisor: An Example	250
26.1.9 Creating a New Graph: An Example	251
26.2 Custom Data Collection	252
26.2.1 Custom.xml	253
26.2.2 Queries	253
26.2.3 Data Collection Attributes	255
26.3 Event Notification Blackout Periods	256
26.3.1 Scripting Blackouts	257

Chapter 13 User Interface

Table of Contents

13.1 Initial Log-In	117
13.2 Setting the Timezone and Locale	118
13.3 Menus and Toolbars	118
13.3.1 Main Menus	118
13.3.2 Status Summary	120

This chapter provides an overview of the MySQL Enterprise Monitor user interface.

13.1 Initial Log-In

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

Figure 13.1 Initial setup for the MySQL Enterprise Monitor User Interface

ORACLE MySQL Enterprise Monitor

Welcome to the MySQL Enterprise Dashboard Setup.

To complete installation and configuration, please complete the form below. You can modify these values later on the Settings page.

Create Manager User

The user and password for the Dashboard administrator. You can optionally add DBA, Read-only or additional Manager users on the Settings page.

Username

Password

Confirm Password

Create Agent User

The user and password for the Agent to login to the dashboard. You'll provide these credentials each time you install an Agent.

Username

Password

Confirm Password

Online Updates

Using a direct internet connection or an HTTP proxy, the Dashboard can occasionally check for MySQL product updates, security alerts, and the status of any open My Oracle Support Service Requests.

☒ Enable automatic checking for online updates

☐ Use HTTP Proxy

Data Purge Behavior

How long collected data should be stored before removing

Remove Historical Data Collection Older Than

4 weeks

Remove Query Analyzer Data Older Than

4 weeks

Complete Setup

☒ Complete Setup

You must perform the following tasks:

- Create a user name and password for the **Manager User**. The manager user is used for the initial session, configuring the systems, and the access control list.
- Create a user name and password for the MySQL Enterprise Monitor Agent. The Agent user credentials are used by every monitoring agent to connect to MySQL Enterprise Service Manager.



Note

It is possible to configure additional Agent users should your system require it. For more information, see [Chapter 23, Access Control](#).

- 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**. If your organisation uses a HTTP proxy, you must check the Use HTTP Proxy field, and complete the fields displayed when this is enabled.

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.

13.2 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.

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, and the built-in Agent is attempting to auto-discover additional MySQL instances on the host.

13.3 Menus and Toolbars

This chapter describes the menus and toolbars of the MySQL Enterprise Monitor user interface.

13.3.1 Main Menus

This section describes the main menus of the user interface.

Dashboards

- **Overview**: opens the **Overview** dashboard. This dashboard provides a high-level view of the current state of your monitored assets. For more information, see [Chapter 14, Overview](#). This is the first page displayed after the initial setup steps are completed.

- **Replication**: opens the **Replication** dashboard. This dashboard provides a detailed view of the current state of your monitored replication servers. For more information, see [Chapter 17, Replication Dashboard](#).
- **MySQL Instances**: opens the **MySQL Instances** dashboard. This dashboard provides a detailed view of the current state of your monitored instances. It also permits the addition, removal, or editing of connections to MySQL instances. For more information, see [Chapter 15, MySQL Instances Dashboard](#).

Events

The **Events** page lists all the events for the monitored assets to which you have access. See [Figure 20.1, “Events Page with Filter”](#) for more information.

Query Analyzer

Opens the Query Analyzer page. See [Chapter 27, Using the Query Analyzer](#) for more information.

Reports and Graphs

The **reports and Graphs** menu contains the following links:

- **All Timeseries**: opens the **All Timeseries** graphs page. See [Section 18.1, “All Timeseries Graphs”](#) for more information.
- **Database File I/O**: opens the Database File I/O page. This page displays details and graphs of latency statistics taken from Performance Schema I/O event data.

This page utilises the `sys` schema and is only supported on MySQL versions 5.6 and 5.7.

- **InnoDB Buffer Pool Usage**: opens the **InnoDB Buffer Pool Usage** block graph. This graph displays a graphical overview of the data stored in the InnoDB Buffer Pool.

See [Section 18.3, “InnoDB Buffer Pool Usage”](#) for more information.

Refresh

Sets the page to refresh automatically according to a schedule. It is also possible to pause the page refresh using the pause button adjacent to the **Refresh** drop-down list.



Note

The pause is temporary. If the page is manually refreshed, the pause is cancelled and the defined refresh behavior resumes.

To resume the defined page refresh, click the button again.

User

This section describes the contents of the User menu.

- **User Preferences**: opens the **User Preferences** page, enabling the user to change their username, full name, password, timezone, and locale.
- **Logout**: ends the current user's session.

Global Settings (Gear)

The Global Settings menu contains links to the following configuration pages:

- **Groups:** opens the **Manage Groups** page.
 - **Advisors:** opens the **Advisors** page. See [Chapter 19, Advisors](#) for more information.
 - **Event Handling:** opens the **Event Handling** page. See [Chapter 20, Events and Event Handling](#) for more information.
 - **Settings:** opens the **Settings** page. See [Chapter 25, Global Settings](#) for more information.
 - **Roles:** opens the **Roles** page. See [Chapter 23, Access Control](#) for more information.
 - **Users:** opens the **Users** page. See [Chapter 23, Access Control](#) for more information.
 - **Diagnostic Report:** generates and downloads the user diagnostic file. This file contains information on the application, property files, stack traces, and all log files.
- This file is intended for MySQL Support, to assist them in diagnosing any issues you may have with your installation.
- For more information, see [Section D.1, “Diagnostics Report”](#).

13.3.2 Status Summary

The Status Summary bar displays the current status of the monitored hosts and instances. Each icon, and its adjacent number, link to pages which provide more detail.

Figure 13.2 Status Summary



The icons, from left to right, represent the following:

- **Hosts Monitored:** the number of successfully monitored hosts. Links to the **MySQL Instances** dashboard.



Note

An agent must be installed on a host to monitor that host. It is not possible to monitor a host without a local agent installed on it. Only MySQL instances can be monitored remotely.

- **MySQL Instances Monitored:** the number of successfully monitored MySQL instances. Links to the **MySQL Instances** dashboard.
- **MySQL Instances with Bad Connection Configurations:** the number of incorrectly configured instance connections.
- **MySQL Instances Unmonitored:** the number of running MySQL instances which are not currently monitored. Links to the **Unmonitored MySQL Instances** list on the **MySQL Instances** dashboard.

See [Unmonitored MySQL Instances](#) for more information.

- **Emergency Events:** the number of current emergency events. Links to the **Events** page and sets the filter to the status **Emergency** and state **Open**.



Note

The hosts and instances represented in the Status Summary depend on the permissions defined for the user. If the user is assigned to a specific group, only the

issues originating from the servers in that group are displayed in the system status bar. For example, the Hosts Monitored icon only displays the total number of hosts in the group assigned to the user's role.

Chapter 14 Overview

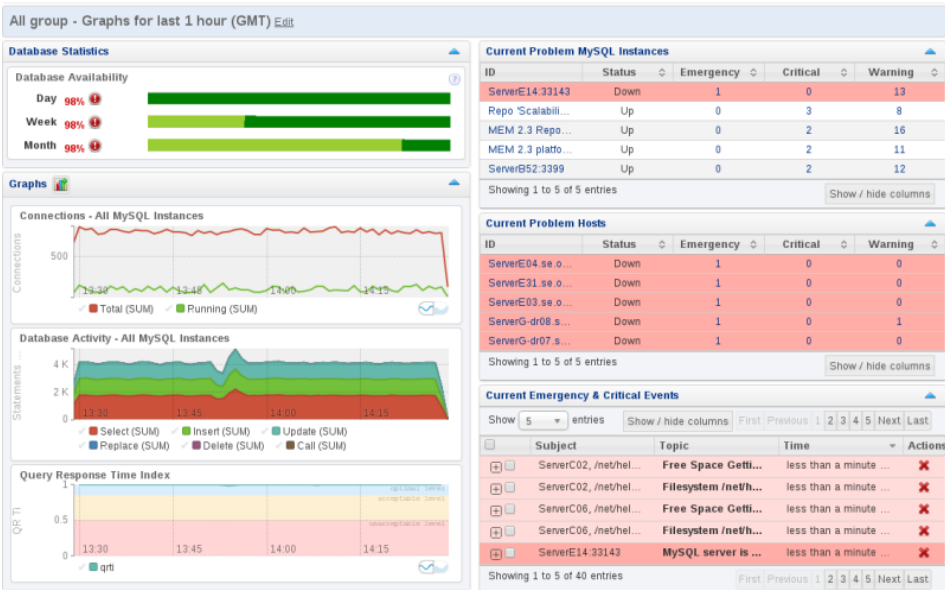
Table of Contents

14.1 Database Statistics	123
14.2 Overview Graphs	124
14.3 General Database Statistics	125
14.4 Group Overview Configuration	125

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.

Figure 14.1 Overview Dashboard



14.1 Database Statistics

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 allows 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 MySQL Enterprise Monitor Agent attempting a connection to the monitored MySQL instances (by default every 1 second, but this is configurable within the Advisor), to check whether the MySQL Instance is actively accepting new connections.

To see a summary of the instance availability per period, hover the cursor over any of the availability bars to display the Availability pop-up. The availability is broken down in to four categories:

- **Available:** The percentage of time the monitored instances were actively monitored.
- **Up:** The instance **Uptime** reports the percentage of time the instance was running but was not monitored.
- **Unreachable:** The percentage of time a monitored MySQL Instance did not respond.
- **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.

14.2 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 the Overview page. To add a graph, do the following:

1. Click the **Add a new graph** button. The graph selection drop-down list is displayed. If the graph is currently displayed on the Overview page, its entry is greyed out in the selection drop-down list.
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.

14.3 General Database Statistics

The remaining Databases Statistics graphs are designed to show a high level picture of the concurrency (Connections - All MySQL Instances), throughput (Database Activity - All MySQL Instances) and response times (Query Response Time Index) of the MySQL Instances within the selected group. These allow you to quickly spot if the profile of activity within the environment has changed.

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 had those 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 are displayed at the top of the list.

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.

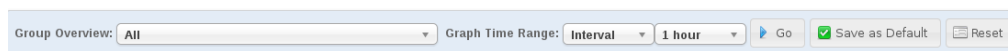
This enables you to see all current high priority events within a single panel for all monitored assets.

14.4 Group Overview Configuration

The **Group Overview** configuration bar enables you to select the 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 14.2 Group Overview Filter Configuration



The controls in the **Group Overview** configuration bar are described in the following table:

Table 14.1 Group Overview Filter Configuration

Name	Description
Group Overview	Drop-down list containing all groups defined in MySQL Enterprise Service Manager.
Graph Time Range	Drop-down lists containing the time periods to apply to the graphs. The possible values are: 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. Graph time ranges apply to the timeseries graphs, only. They do not apply to the Availability, Problem, or Current Emergency information.

Name	Description
Go	Applies the new configuration to the Overview dashboard.
Save as Default	Sets the selected group and time range as the default. It is not possible to save a date range, using From/To, as the default for a group.
Reset to Default	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.

Chapter 15 MySQL Instances Dashboard

Table of Contents

15.1 MySQL Instance Dashboard UI	127
15.2 MySQL Instance Details	129
15.3 Adding Instances	131
15.3.1 Adding a MySQL Instance	131
15.3.2 Adding Multiple MySQL Instances	135
15.4 Filtering MySQL Instances	135

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 15.2, “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](#).

15.1 MySQL Instance Dashboard UI

This section describes the MySQL Instances Dashboard.

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.



Note

These buttons depend on the permissions defined for the user. If the permission **MySQL Instances** is set to Read-Only, the buttons are visible, but inactive, and it is not possible to open the associated lists.

- **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 15.1 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 15.2 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 15.3.1, “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 15.3 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.

15.2 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 15.3.1, “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 15.4 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.
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



Note

The menu items listed in this section depend on the permissions defined. If you do not have the required permissions, some or all of these menu items may be inactive.

The group-level context menu contains the following:

- **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.

The instance-level menu contains the following:

- **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. 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**

Generating a diagnostic report is an expensive operation, the **Diagnostics Report** 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 triggers generation of a new report.

- **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.

15.3 Adding Instances

This section describes how to add MySQL Instances to MySQL Enterprise Monitor. The following topics are described:

- [Section 15.3.1, “Adding a MySQL Instance”](#)
- [Section 15.3.2, “Adding Multiple MySQL Instances”](#)

**Note**

These buttons depend on the permissions defined for the user. If the permission **MySQL Instances** is set to Read-Only, the buttons are visible, but inactive, and it is not possible to add instances.

To add instances, the permission **MySQL Instances** must be set to Administer.

See [Chapter 23, Access Control](#) for more information.

15.3.1 Adding a MySQL Instance

This section describes how to monitor a MySQL instance.

Connection Settings

The Connection Settings tab

Figure 15.1 Add Instance Connection Settings

Connection Settings | Encryption Settings | Advanced Settings | Group Settings

Monitor From 
 dfunkier

Connect Using 
 TCP/IP

Instance Address  Port 
 Port: 3306

Admin User  Admin Password 
 Admin User: root

Auto-Create Less Privileged Users 
 Yes

General User  General Password 
 General User:
 General Password:

Limited User  Limited Password 
 Limited User:
 Limited Password:

Add Instance Cancel

Table 15.5 Connection Settings Tab

Name	Description
Monitor From	Select an Agent from the list of Agents to monitor this MySQL database Instance. It is recommend installing one Agent per Host and using that Agent to monitor all its MySQL Instances. 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.  Note If the instance is monitored remotely, it is not possible to retrieve any information on the host. To monitor a host, an agent must be installed on that host.
Connect Using	Select TCP/IP or a socket to connect to the instance. Socket connections can only be used for an Agent that is installed on the same machine as the target instance, and do not work with instances running on Windows.

Name	Description
Instance Address and Port	The IP address, or valid hostname, and port number the instance is listening on. If the host/agent chosen is local to this instance, you should use 127.0.0.1 here.
Admin User and Password	The root user, or user with the privileges defined in Creating the Admin User and the password.
Auto-Create Less Privileged Users	Choose Yes to create the General and Limited users on the MySQL instance. You must add a user name and password for both. For more information on these users, see Section 5.2, “Creating MySQL User Accounts for the Monitor Agent”. Choose No if you intend to use the Admin user for all data collection.  Note It is strongly recommend to use the General and Limited user.
General User and Password	Add a user name and password for the General User.
Limited User and Password	Add a user name and password for the Limited User.

Encryption Settings

Figure 15.2 Add Instance Encryption Settings

Connection Settings
Encryption Settings
Advanced Settings
Group Settings

Require Encryption ?

No

Allow Self-Signed Certificates ?

No

CA Certificate ?

Add Instance

Cancel

Table 15.6 Encryption Settings Tab

Name	Description
Require Encryption	Defines whether the connection uses TLS for security.
Allow Self-Signed Certificates	Specifies whether the connection permits self-signed certificates.
CA Certificate	Paste the CA certificate's contents here.  Note This is not required if you are using a self-signed certificate.

Advanced Settings

Figure 15.3 Add Instance Advanced Settings

Connection Settings | Encryption Settings | **Advanced Settings** | Group Settings

Discover Replication Topologies ?

Yes ▾

MySQL Instance Identity Source ?

Default ▾

Inventory Table Schema ?

mysql

Connection Timeout ?

10000

Socket Timeout ?

60000

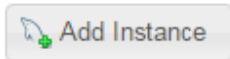



Table 15.7 Advanced Settings

Name	Description
Discover Replication Topologies	Specifies whether the agent attempts to discover if the instance is part of a replication group and discover the other members of that replication group. When performing replication topology discovery, the agent attempts to read the slave's master.info, and use the stored credentials to log in to the master to read its inventory table and retrieve the master's UUID. If set to No, no replication discovery is attempted.
MySQL Identity Source	Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.

Name	Description
	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.  Note Host Plus Datadir can be used only if the agent is running on the same host as the MySQL instance for this connection.
Inventory Table Schema	When the Agent connects to the MySQL Instance, it creates an inventory table, if one does not already exist, and stores two rows within it: a generated Instance UUID, and the host ID. By default this is created within the <code>mysql</code> database. On shared hosts or cloud environments this may not be accessible to the Agent user; provide a database name to override where the inventory table is created.
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.



Note

To add groups, you must have the **Server Group** permission set to **Read-Only**, at least.

To create new groups, you must have the **New Group Creation** permission set to **Administer**.

15.3.2 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.

15.4 Filtering MySQL Instances

To search for specific instances, click the filter icon. The **MySQL Instance Details** filter is displayed.

Figure 15.4 MySQL Instance Filter

The screenshot shows a window titled "MySQL Instance Details" with a filter section. The filter section includes the following fields:

- Server Name**: Contains dropdown, Value input box.
- Server UUID**: Contains dropdown, Value input box.
- Server ID**: Contains dropdown, Value input box.
- Query Analyzer**: (any) dropdown.
- MySQL Version**: Input box.
- Agent Version**: Input box.
- Operating System**: Contains dropdown, Value input box.

At the bottom of the filter section are two buttons: "Filter" and "Reset".

Table 15.8 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 server_id .
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.

**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 16 Managing Groups of Instances

This chapter describes groups.

Groups enable you to place instances into useful collections. For example, you can create groups for development and production instances. Instances added to each group inherit the Advisors scheduled for that group.

The primary uses for groups are:

- **Access Control:** You can assign users to specific groups. This means the user sees only those instances in the group to which they have rights. The groups are associated with Roles, and the users are assigned to the roles. For more information, see [Chapter 23, Access Control](#).
- **Replication:** MySQL Enterprise Monitor automatically creates groups for replication topologies. That is, if a master-slave(s) relationship is detected, the relevant group is created to contain all members of that topology.
- **Convenience:** grouping related instances together, in order to ensure consistent Advisor scheduling and event generation.

Creating Groups



Important

To create groups, the user must be assigned to a role with the **New Group Creation** permission set to Administer. To view groups, they must have the **Server Group** permission set to at least Read-Only.

To open the Groups page, click the Groups link on the Settings menu.

Figure 16.1 Group Management Page

To create a group, do the following:

1. Click **Create**. The **Create Group** frame is activated.
2. Define a **Group Name** and a Description.
3. It is possible to create empty groups, and add the instances later, or to allow the Agent installations to add the instances to the groups by adding the group name to the **Monitor Group** field in the installer.
4. To add instances to the group, select the **Assets** tab.
5. Select the required instances by checking the checkbox beside the instance.

6. Click **Save** to save your new group. Click Cancel to discard your changes.
7. To edit a group, select it in the list, and edit as required.



Note

Editing groups requires the user be assigned to a Role with the **Server Group** and **MySQL Instances** permissions set to Administer.

Deleting Groups

To delete a group, you must have be a member of a role with the **Server Group** permission set to Administer.

To delete a group, select the group in the groups list and click **Delete**.

Chapter 17 Replication Dashboard

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 **Groups** page in the same way as other groups. For more information, see [Chapter 16, Managing Groups of Instances](#). However, any slaves removed from a server group are automatically restored to that group. You can also add non-slaves to a replication grouping.

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](#).

Chapter 18 Reports and Graphs

Table of Contents

18.1 All Timeseries Graphs	143
18.1.1 Graph Controls	143
18.1.2 Graph Types	145
18.2 Database File I/O and Lock Waits	145
18.2.1 sys Schema	145
18.2.2 Database File I/O Graphs and Reports	146
18.2.3 Lock Waits Report	148
18.3 InnoDB Buffer Pool Usage	148

This chapter describes the Reports and Graphs available in MySQL Enterprise Monitor.

18.1 All Timeseries Graphs

This section describes the **All Timeseries Graphs** page.

18.1.1 Graph Controls

This section describes the controls available on the **All Timeseries Graphs** page.

Asset Selector

The Asset Selector is a treeview container on the left side of the page. It lists all the hosts to which the user has access. The hosts are organised by group.

The group displayed depend on those configured. If no groups are configured, the All group is the only group displayed, and all hosts are contained within it.

The graphs displayed depend on what is selected in the Asset Selector. The Asset Selector displays the following:

- Group
- Host
- Instance
- Network Interface
- Filesystem
- Agent (if installed on host)



Important

The Asset Selector displays only those assets to which you have access.

The Asset Selector can be filtered to display specific assets. To filter the Asset Selector, click the filter icon in the top-right corner of the Asset Selector frame. The **Filter** field is displayed. Enter the text to filter on. The Asset Selector displays the filtered results.

To display monitored instances only, deselect the **Show All Assets** checkbox.

Graph Filter

The graph filter enables you to display a subset of the available graphs.

Table 18.1 Timeseries Graph Filter

Name	Description
Graph Name	Opens a drop-down menu listing the available search types: • Contains • Doesn't Contain • Regex • Negative Regex
Value	Free text field for the search term or regular expression.
Time Range	Drop-down lists containing the time periods to apply to the graphs. The possible values are: • 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. <pre><xi:include href="mem-ui-database-file-io.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/> <xi:include href="mem-innodb-bufferpool-report.xml" xmlns:xi="http://www.w3.org/2001/XInclude"/></pre>
Filter	Apply the defined filter.
Save as Default	Sets the selected group and time range as the default. It is not possible to save a date range, using From/To, as the default for a group.
Reset to Default	Resets the graph display to the previously saved values.

Graph Manipulation

This section describes the various actions you can perform on individual graphs.

- **Graph Height**: slider which enables you to increase or decrease the height of the graph in pixels. This slider does not affect the values of the x or y axes, just resizes the graph.
- **Export as CSV**: downloads a CSV containing all data currently displayed in the selected graph.
- **Export as PNG**: generates a PNG image file of the selected graph. The image is displayed in a pop-up. To save the image, right-click and select **Save image as....**
- **Move**: enables you to move the selected graph to another location on the page.
- **Stacked/Line**: enables you to change how the graph is displayed. **Line** displays a line graph, while **Stacked** displays each data source as a solid color.
- **Legend**: lists the sources of information displayed in the graph. The color of the name matches the line/stack used in the graph. To display individual sources, click the required source in the **Legend**. To highlight individual sources in the graph, hover the cursor over the source's name.

Graph Query Analysis

Graph Query Analysis enables you to examine the queries which were running during specific intervals. To open the Query Analyzer for a specific range on a graph, do the following:

1. On a graph, select a range by clicking at the required start point, and dragging the cursor across the graph until you reach the required interval endpoint and release the mouse button. This selects the range.
2. Two icons are displayed in the top-right corner of the selection. An x to close the selection, and a database icon. Click the database icon to open the Query Analyzer's **Browse Queries** page.

Browse Queries displays all the queries which were running during the defined time period. This enables you to drill down into potential query bottlenecks and performance hotspots and tune your queries accordingly.

For more information on the Query Analyzer, see [Section 27.3, “Query Analyzer User Interface”](#).

18.1.2 Graph Types

The following are the graph types:

- **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. Such as the group-level **Database Activity - All MySQL Instances** graph.
- **Treemap:** A 2D hierarchical proportional-representation graph. See [Section 18.3, “InnoDB Buffer Pool Usage”](#) for an example.

18.2 Database File I/O and Lock Waits

This chapter describes the Database File I/O and Lock Waits reports. These reports identify I/O hot spots and lock wait contention in your application using the [sys](#) schema, thereby enabling you to tune the performance of your queries.



Important

The Database File I/O requires the MySQL [sys](#) schema, which is supported on MySQL 5.6 and 5.7, only.

18.2.1 sys Schema

The [sys](#) schema is a set of views, stored procedures, and functions, which provide access to the instrumentation data of the Performance Schema.

The [sys](#) schema is installed by default in MySQL 5.7, but must be installed manually in earlier versions of MySQL.

On the Database File I/O and Lock Waits pages, if a compatible MySQL instance is selected, but `sys` schema is not installed, MySQL Enterprise Monitor prompts you to install it. To install `sys` schema, click **Install MySQL sys schema**. If the selected instance is incompatible, a message is displayed informing you that it is not possible to run these reports against the selected schema.



Important

If your instance already contains a schema named `sys`, you must rename it before installing MySQL `sys` schema.

For information on how to install `sys` schema from the command line, see the installation instructions within the github repository: [sys schema on GitHub](#).

18.2.2 Database File I/O Graphs and Reports

This section describes the **Database File I/O** reports and graphs.

Each tab contains the following common elements:

- **Show n Entries:** Number of entries to show per page.
- **Search:** search the contents of the page.
- **Show/Hide Columns:** enables you to change the column set displayed on the page by selecting or deselecting the columns.
- **Page Navigation:** buttons enabling you to navigate the pages of the report.

For more information on the data retrieved in these reports, see the XREFTO REFMAN.

I/O By File

Shows the top global I/O consumers by latency, and by file. The data is retrieved from `sys.x $io_global_by_file_by_latency`, and sorted by total latency by default.

Figure 18.1 Database File I/O By File

Database File I/O

By FileBy Wait TypeBy Thread

@@datadir: /export/home/etools/mysql-servers/data/mysql-3357/
@@basedir: /export/home/etools/mysql-servers/mysql-3357/
@@tmpdir: /tmp/Refreshed: Feb 19, 2016 5:07:40 pmReload

Show10entriesSearch:Show / hide columnsFirstPrevious12345NextLast

File	Total I/Os	Total I/O Latency	Read I/Os	Read I/O Latency	Write I/Os	Write I/O Latency	Misc I/Os	Misc I/O Latency
@@datadir/ib_logfile0	13.08 K	1.78 m	7	9.23 ms	6.66 K	284.33 ms	6.41 K	1.78 m
@@datadir/ibdata1	6.12 K	35.6 s	332	288.3 ms	3.19 K	20.21 s	2.6 K	15.1 s
@@datadir/ibtmp1	181.9 K	10.43 s	0	0 ps	181.89 K	10.43 s	4	257.3 us
@@datadir/platformtest/proxytest.ibd	2.97 K	8.39 s	293	260.45 ms	1.4 K	30.79 ms	1.27 K	8.1 s
@@datadir/mysql/user.MYD	201.15 K	3.52 s	51.45 K	3.34 s	164	2.17 ms	149.53 K	168.88 ms
@@datadir/mysql/tables_priv.MYD	26.83 K	326.4 ms	26.16 K	318.23 ms	666	8.02 ms	8	152.57 us
@@datadir/mysql/help_topic.ibd	65	316.83 ms	60	316.79 ms	0	0 ps	5	42.47 us
@@datadir/mysql/db.MYD	6.53 K	226.48 ms	6.52 K	226.31 ms	6	78.49 us	11	93.89 us
@@datadir_em_9572/vitro04_20160209_110141_3357_a1.frm	41	161.15 ms	7	12.89 us	21	44.43 us	13	161.09 ms
@@datadir_em_9572/vitro04_20160209_110141_54044_a1.frm	41	105.94 ms	7	15.83 us	21	35.59 us	13	105.89 ms

Showing 1 to 10 of 338 entriesFirstPrevious12345NextLast

I/O By Wait Type

Shows the top global I/O consumers by latency. The data is retrieved from `sys.x $io_global_by_wait_by_latency`, and sorted by total latency, by default.

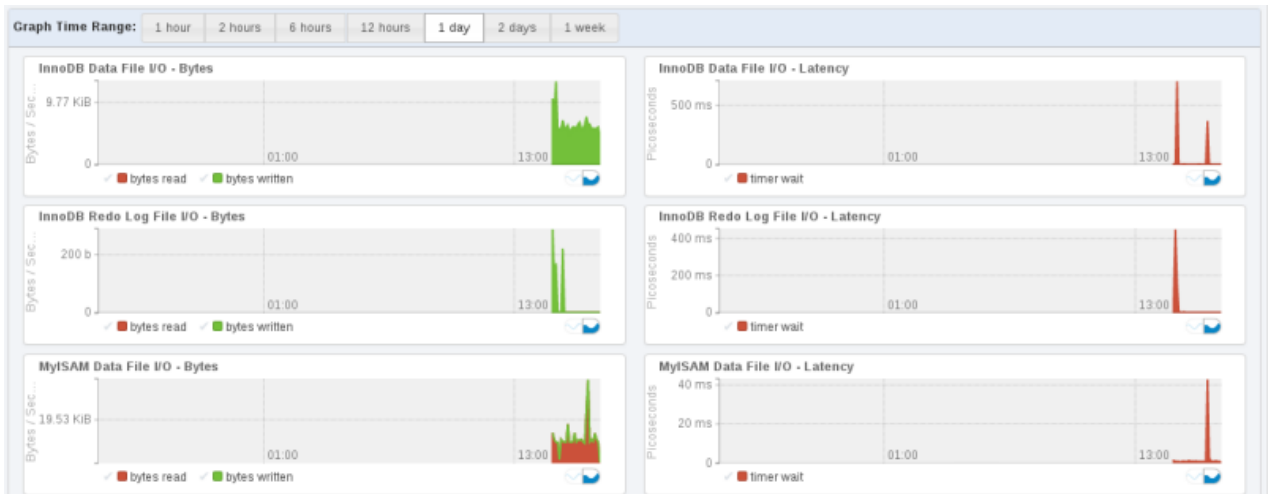
This report is a combination of report and graphs. The graphs can be redrawn based on a time range. To change the time range, select one of the range buttons. Values range from 1 hour to 1 week.

Figure 18.2 Database File I/O By Wait Type Report

Database File I/O												
By File By Wait Type By Thread			Refreshed: Feb 19, 2016 5:07:33 pm Reload									
Show 10 entries			Search:		Show / hide columns		First Previous 1 2 Next Last					
I/O Type	Total I/Os	Total I/O Latency	Avg I/O Latency	Max I/O Latency	Read I/Os	Read I/O Latency	Total Read	Avg Read	Write I/Os	Write I/O Latency	Total Written	Avg Write
innodb/innodb_log_file	13.08 K	1.78 m	8.17 ms	1.6 s	7	9.23 ms	68.5 KIB	9.79 KIB	6.66 K	284.33 ms	5.17 MiB	5.17 MiB
innodb/innodb_data_file	192.36 K	55.12 s	286.57 us	891.07 ms	776	1.13 s	14.14 MiB	18.66 KIB	186.97 K	30.75 s	2.91 GiB	2.91 GiB
sql/FRM	719.28 K	4.83 s	6.71 us	160.83 ms	1.29 K	65.12 ms	310.6 KIB	246 b	1.58 K	3.12 ms	322.63 KIB	322.63 KIB
myisam/dfile	236.96 K	4.19 s	17.7 us	136.02 ms	85.76 K	4.01 s	2.85 GiB	34.87 KIB	836	10.27 ms	608.15 KIB	608.15 KIB
myisam/kfile	880	150.53 ms	171.06 us	32.07 ms	20	32.09 ms	5.17 KIB	264 b	839	118.28 ms	131.29 KIB	131.29 KIB
sql/dbopt	24.47 K	93.29 ms	3.81 us	963.31 us	0	0 ps	0 b	0 b	0	0 ps	0 b	0 b
sql/ERRMSG	5	583.24 us	116.65 us	514.36 us	3	64.48 us	73.28 KIB	24.43 KIB	0	0 ps	0 b	0 b
archive/data	72	330.9 us	4.6 us	6.81 us	0	0 ps	0 b	0 b	0	0 ps	0 b	0 b
sql/casetest	10	184.41 us	18.44 us	94.59 us	0	0 ps	0 b	0 b	0	0 ps	0 b	0 b
sql/misc	12	112.92 us	9.41 us	15.59 us	0	0 ps	0 b	0 b	0	0 ps	0 b	0 b
Showing 1 to 10 of 17 entries												
										First Previous 1 2 Next Last		

The following shows a subset of the graphs available on the I/O By Wait Type tab:

Figure 18.3 Database File I/O By Wait Type Graphs



I/O By Thread

Shows the top I/O consumers by thread, ordered by total latency. The data is retrieved from `sys.x $io_by_thread_by_latency`, and sorted by latency, by default.

Figure 18.4 Database File I/O By Thread

Database File I/O

By File By Wait Type By Thread

Refreshed: Feb 19, 2016 5:07:40 pm [Reload](#)

Show 10 entries Search: Show / hide columns First Previous 1 2 Next Last

Connection ID	Account	Total I/Os	Total I/O Latency	Avg I/O Latency	Max I/O Latency
null	srv_master_thread	8.8 K	1.22 m	54.12 ms	1.6 s
null	page_cleaner_thread	188.06 K	46.15 s	140.69 ms	891.07 ms
null	io_log_thread	2.44 K	32.48 s	13.34 ms	1.24 s
null	io_write_thread	1.92 K	7.79 s	4.05 ms	577.67 ms
null	io_write_thread	701	945.79 ms	1.35 ms	535.42 ms
null	buf_dump_thread	205	515.86 ms	2.52 ms	47.61 ms
null	main	2.4 K	451.35 ms	121.55 us	17.98 ms
582950	root@localhost	51.42 K	228.88 ms	3.55 us	41.55 us
1429877	agent_limited@localhost	36.13 K	177.76 ms	3.4 us	54.81 us
3003449	l1002ty72_214762@localhost	7.04 K	31.66 ms	9.06 us	101.04 us

Showing 1 to 10 of 19 entries First Previous 1 2 Next Last

18.2.3 Lock Waits Report

To open the Lock Waits reports, select **Lock Waits** from the **Reports & Graphs** menu.

InnoDB Row Lock Waits

This report retrieves data on InnoDB row locks from `sys.x$innodb_lock_waits`.

Table Metadata Lock Waits



Important

Table Metadata Lock Waits is supported on MySQL 5.7 only. This report relies on instrumentation introduced in MySQL 5.7.

This report retrieves data on MySQL 5.7 table metadata locks from `sys.x$schema_table_lock_waits`.

18.3 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 19 Advisors

Table of Contents

19.1 Advisors Page	151
19.2 Advisor Types	155
19.3 Advisor Thresholds	156
19.4 Advisor Schedules	158

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:

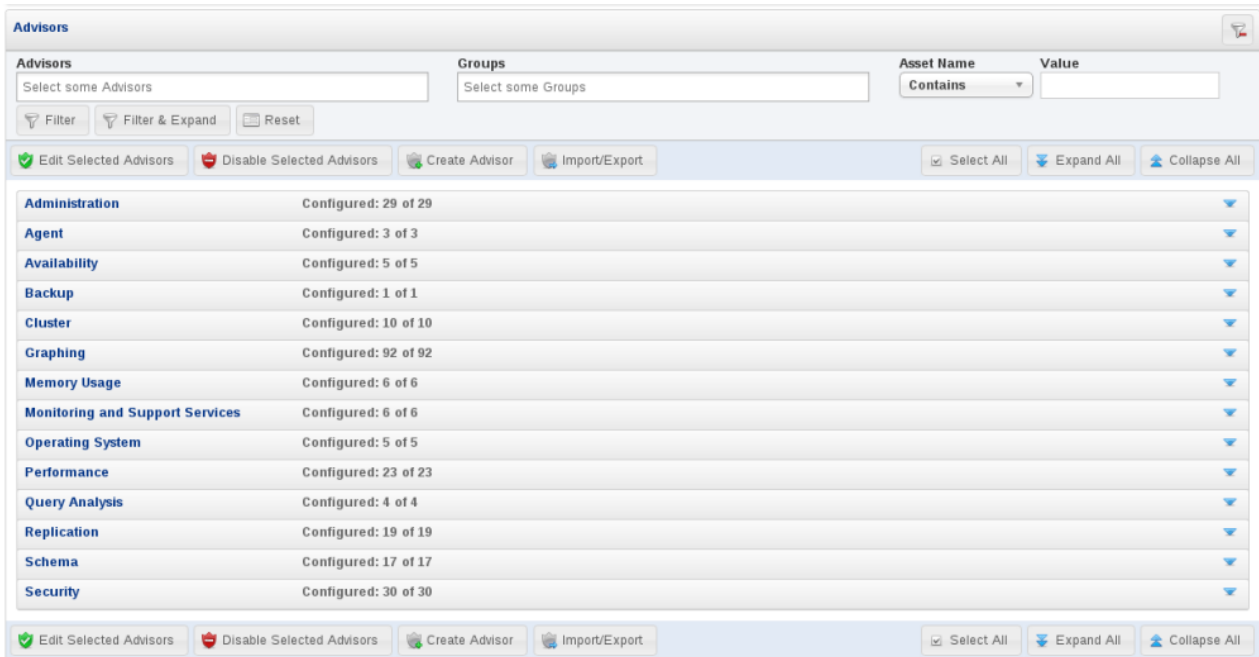
- [Section 19.1, “Advisors Page”](#)
- [Section 19.2, “Advisor Types”](#)
- [Section 19.3, “Advisor Thresholds”](#)
- [Section 19.4, “Advisor Schedules”](#)

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 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 22, GUI-Based Advisor Reference](#) and [Chapter 21, 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:

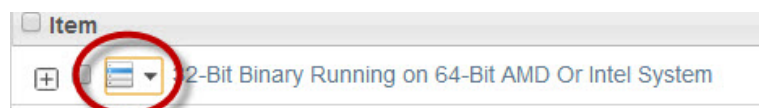
Table 19.2 Advisor Information Listing

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 20, Events and 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 Pop-up Menu

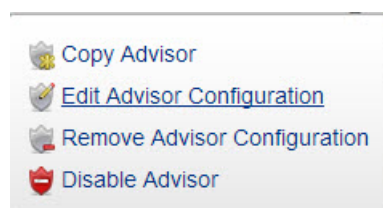


Table 19.3 Advisor Edit Menu Controls

Name	Description
Copy Advisor	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.  Note This option is only available for expression-based Advisors.
Edit Advisor Configuration	Opens the Edit Advisor dialog. This enables you to change the parameters and schedule of the selected advisor.
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	Deletes the selected advisor.  Note Only available for custom Advisors. It is not possible to delete the default Advisors.

Group and Host Menu

Each advisor contains the list of all groups defined in MySQL Enterprise Monitor. To see these groups, expand the contents of the Advisor by clicking on the Advisor's name. This enables you to specify the Advisors you want to run for each group.

The top-level advisor contains the global configuration for all groups. That is, the configuration at the advisor-level applies to all groups and hosts it contains. Each nested group, and the monitored hosts contained in the group, have a drop-down menu enabling you to override the global configuration for each group or host, or disable the advisor for the specific group or host. Any change in advisor configuration at the group or host level, overrides the global configuration specified at the advisor level.

To open the Group menu, expand the Advisor and select the drop-down icon next to the Group name. The 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 Pop-up Menu

Table 19.4 Advisor Filter Controls

Name	Description
Advisors	Opens a drop-down menu listing all available Advisors. You can select multiple Advisors.
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: • Contains • Doesn't Contain • Regex • Negative Regex
Value	Free text field for the search term or regular expression.
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 the 'General' configuration window for the Agent Health Advisor. It contains the following settings:

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

Parameters Schedule 1 1 0 P

Thresholds ?

☒ Notice Threshold

75

☒ Warning Threshold

85

☒ Critical Threshold

95

☒ Emergency Threshold

100

☒ Save ☐ Cancel

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:

- **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.
- **Daily:** collects data at the defined time. This is useful for collections with a large overhead on the monitored instance, enabling you to schedule the collection for an off-peak time.
- **Disabled:** deactivates the advisor for all monitored assets, or for the selected group or host.

Chapter 20 Events and Event Handling

Table of Contents

20.1 Events	159
20.2 Event Handling	163
20.2.1 Event Handlers	163
20.2.2 Event Handling Page	163
20.3 Creating Event Handlers	168
20.3.1 Event Action Log	170
20.3.2 Suspending an Event Handler	170

This chapter describes Events and Event Handlers.

Events are displayed if an Advisor Threshold is crossed, and are used to inform you of errors or potential problems with your implementation.

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.

20.1 Events

Advisors generate events if one, or more, of the defined thresholds are crossed by the monitored value.

Events are displayed on the **Events** page. Emergency and Critical events also appear 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 20, Events and Event Handling](#).

To view open events, click the **Events** link. The tree-view Asset Selector on the left enables you to choose which group's or asset's events are displayed.



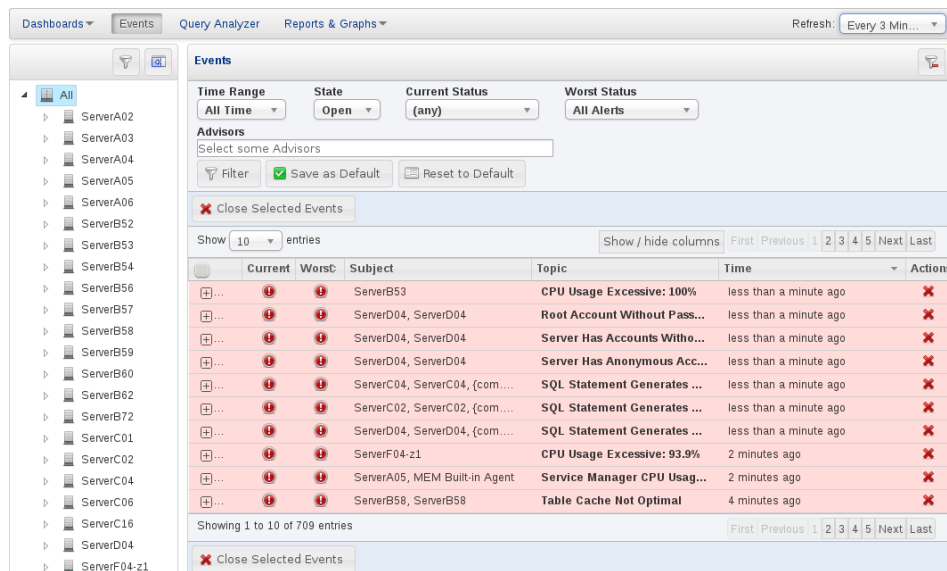
Important

The events displayed are dependent on the permission of the role to which you are assigned. If you are a member of a group-specific role, the events displayed are generated by the members of that group, only. Your ability to close events is also governed by the permissions of your role.

For more information, see [Chapter 23, Access Control](#).

The following image shows the **Events** page with filter enabled:

Figure 20.1 Events Page with Filter



Event Filter

The Event filter enables you to filter your events based on criteria.

Table 20.1 Events Filter Controls

Name	Description
Time Range	Enables you to choose a time range. Possible values are: • All Time: filter on all events. • Interval: displays the Interval drop-down list, enabling you to filter on a defined range from the current time. For example, if you select 15 minutes, the filter returns events generated in the last 15 minutes, only. • From/To: displays From and To fields enabling you to define a date and time range to filter on.
State:	Enables you to choose the events states to filter on. The available choices are Any , Open , or Closed .
Current Status	Enables you to filter on specific current status.
Worst Status	Enables you to filter on specific worst status.
Advisors	Enables you to filter events based on the Advisors which generated them.
Filter	Click to filter the events list on the defined criteria.
Save as Default	Saves the current filter as the default view of the events list.
Reset to Default	Reverts any changes, and restores the saved default criteria.

Events List

The Events list displays all events for the selected group or asset.

Table 20.2 Events List Columns

Name	Description
Multi-select checkbox	Enables you to select all events.
Current	Displays an icon indicating the current status of the event.
Worst	Displays an icon indicating the worst status of the event.
Subject	Displays the hostname and location of the problem. For example, if the issue is low disk space on a monitored host, the Subject displays the hostname and the path to the drive which is running out of space. If the issue is related to an instance running on that host, the hostname, and the name and port number of the instance are displayed.
Topic	Displays the short description of the event.
Time	Displays the approximate time the event was generated.
Actions	Displays the possible actions. Click X to open the Close Events dialog.

The Event severities are:

- **Emergency:** The event is an emergency and requires immediate attention.
- **Critical:** The event is critical and requires immediate attention. Critical events indicate that a serious problem has occurred or is about to occur.
- **Warning:** The event is something to investigate and schedule for correction, but does not immediately affect the operation of your server, such as free space on a disk, or a table cache is inadequately sized.
- **Notice:** The event is for informational purposes. Notice events call attention to issues that do not affect the operation of your server, such as minor configuration issue.
- **Success:** The rule executed successfully with no issues. It also indicates an event, previously in a Critical or Failure state, has returned to normal.
- **Unknown:** The current status of the event/rule cannot be determined.
- **Closed:** The issue has been corrected and marked closed.

Icon	Description
	Red and orange flame icon indicates an emergency alert.
	Round red icon indicates a critical alert.
	Triangular yellow icon indicates a warning.
	Blue octagon with the letter "i" indicates an informational alert.
	Green check indicates that the Advisor ran successfully and no alert was generated.
	Skull icon indicates that the status of the Advisor is unknown.

Closing Events

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 20, Events and Event Handling](#).



Important

Not all Advisors generate events which can be auto-closed. See [Chapter 21, Expression-Based Advisor Reference](#) and [Chapter 22, 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 selecting the checkbox **After closing, take me to the page for adjusting schedules of Advisor(s) that reported these events**. This option opens the **Advisors** page and selects the relevant Advisors.

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 replaced by 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 data purge settings. For more information, see [Section 25.4, "Data Purge Behavior"](#)

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.

20.2 Event Handling

This section describes the Event Handling of MySQL Enterprise Service Manager.

20.2.1 Event Handlers

Event handlers are conditions associated with actions. If the condition is met, the action is performed.

Event handler conditions are comprised of the following elements:

- Groups of assets or individual assets.



Important

It is not currently possible to select both groups and individual assets, you must select one or the other.

- Advisors you want to raise notifications for.
- Event statuses to trigger the notifications (WARNING, CRITICAL, EMERGENCY, and so on).

The conditions are constructed in the following way:

```
Group AND Advisor AND Status
```

while the contents of the elements are OR clauses. For example:

```
(Group A OR Group B) AND
(Advisor= MySQL Process OR Advisor=CPU Utilization Advisor) AND
(status=Warning OR status=Critical)
```

If the MySQL Process advisor generates a Warning event for one of the contents of Group A, the condition is true and the associated action is triggered. The action can be one of the following:

- Send an email or SNMP notification if one of the following occurs:
 - The condition evaluates as True.
 - The condition evaluates as True and the status changes to any other status.
 - The condition evaluates as True and the status escalates.
- Auto-close the event if the current status of the event is OK, but the prior status matched one of those defined in the condition.

20.2.2 Event Handling Page

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:

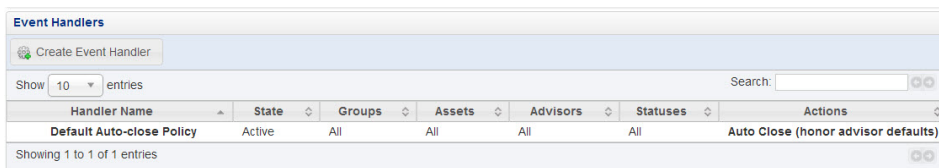
- **Event Handlers:** Lists the event handlers defined on the system. The Default Auto Close Policy is present by default and cannot be edited.
- **Email Notification Groups:** lists the email notification groups defined on the system.

- **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.

20.2.2.1 Event Handlers List

The **Event Handlers** section lists all event handlers defined on the system and enables you to create Event Handlers.

Figure 20.2 Event Handlers section



Event Handlers section contains the following controls:

Table 20.3 Event Handler List Controls

Name	Description
Create Event Handler	Opens the Create Event Handler dialog. For more information, see Section 20.3, “Creating Event Handlers” .
Show * Entries	Select the maximum number of event handlers to display.
Handler Name	Lists the names of the event handlers.
State	Lists the state of the event handler. Possible states are: • 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.

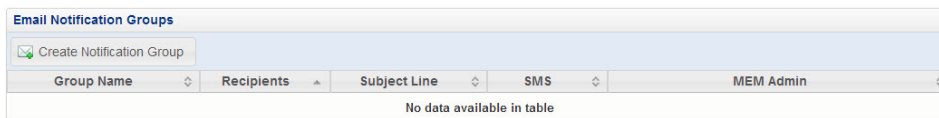
**Important**

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.

20.2.2.2 Email Notification Group Controls

This section describes the controls on the **Email Notification Group** section.

Figure 20.3 Email Notification Groups section



The **Email Notification Groups** contains the following controls:

Table 20.4 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.
Subject Line	The subject line of the notification emails.
SMS	Status of SMS encoding. The following values are possible: • 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: • 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 20.4 Create Group Dialog

Create Group

Group Name

Recipients (comma separated or one email address per line)

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

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

☐ SMS (use SMS encoding for this notification group)

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

Save Notification Group Cancel

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.

20.2.2.3 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 20.5 Email Settings section

Email Settings

☒ Enable Email Notifications

From Address
email@domain.com

SMTP Server
smtp.domain.com

SMTP Server Login

☐ Update Password On Save

☒ Disable JavaMail TLS/SSL

On Save, Send Test Email Message to (optional)

Save Email Settings

The **Email Settings** pane contains the following controls:

Table 20.5 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.

20.2.2.4 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.

20.2.2.5 SNMP Settings

The **SNMP Settings** section enables you to define the SNMP trap configuration, such as SNMP version, SNMP targets, and so on.

Figure 20.6 SNMP Settings section

The **SNMP Settings** pane contains the following controls:

Table 20.6 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 public .

Name	Description
Use the remote MySQL agent host IP address as the SNMP trap agent address for Advisor traps (optional)	Defines the source IP address included in the trap. • 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.

20.2.2.6 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.

20.3 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.

Table 20.7 Create Event Handler Controls

Name	Description
Event Handler Name	Specify a name which uniquely identifies the new event handler.
Filters	
Assets	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.

Name	Description
	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.  Note 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.
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.
Event Handling	
SMTP Notification Groups	Select the groups you want to notify.
SMTP/SNMP Notification Policy	Select one of the following: • 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.
SMTP Rate Limit	Defines the maximum number of SMTP notifications which can be sent per minute. The default is 10.
Send SNMP Traps	Defines whether SNMP Traps are used for notifications.
Auto-Close Events	Defines whether the events are closed after the trigger status changes. The following values are possible: • 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.



Important

Do not define both Assets and Groups, use one or the other.

20.3.1 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.

20.3.2 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.



Note

The rights to suspend event handlers depend on the Event Blackout permission. If this right is not granted to your role, it is not possible to suspend an event handler.

Chapter 21 Expression-Based Advisor Reference

Table of Contents

21.1 Administration Advisors	171
21.2 Agent Advisors	177
21.3 Availability Advisors	178
21.4 Cluster Advisors	179
21.5 Memory Usage Advisors	181
21.6 Monitoring and Support Services Advisors	183
21.7 Operating System Advisors	184
21.8 Performance Advisors	184
21.9 Replication Advisors	190
21.10 Schema Advisors	195
21.11 Security Advisors	199

This chapter describes the MySQL Enterprise Monitor expression-based Advisors.

21.1 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 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 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 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 will run when requested. Unlike triggers, events are not linked to specific table operations, but to dates and times. Using the event scheduler, the database administrator can perform recurring events with minimal hassle. Common uses are the cleanup of obsolete data, the creation of summary tables for statistics, and monitoring of server performance and usage.

Default frequency 00:05:00

Default auto-close enabled 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 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) will read 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 will raise error conditions in certain cases, rather than issue a warning and process the specified command (perhaps with some unintended defaults). This is analogous to MySQL's `sql_mode`, which controls what SQL syntax MySQL will accept, and determines whether it will silently ignore errors, or validate input syntax and data values.

Using the new clauses and settings for `ROW_FORMAT` and `KEY_BLOCK_SIZE` on `CREATE TABLE` and `ALTER TABLE` commands and the `CREATE INDEX` command can be confusing when not running in strict mode. Unless you run in strict mode, InnoDB will ignore certain syntax errors and will create the table or index, with only a warning in the message log. However if InnoDB strict mode is on, such errors will generate an immediate error and the table or index will not be created, thus saving time by catching the error at the time the command is issued.

Default frequency 12:00:00

Default auto-close enabled 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 will occur and your application may experience problems.

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

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, then a table will be "Marked as crashed" if it becomes corrupt, and no sessions will be able to SELECT from it, or perform any sort of DML against it.

Default frequency 06:00:00

Default auto-close enabled 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 will work to open and close tables as needed, however you should avoid having the table cache set too low, causing MySQL to constantly open and close tables to satisfy object access.

If the table cache limit has been exceeded by the number of tables opened in the first three hours of service, then the table cache size is likely set too low.

Default frequency 00:30:00

Default auto-close enabled 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 you will not get valuable information about aborted connections and various other communication errors. This is especially important if you use replication so you get more information about what is happening, such as messages about network failures and reconnection.

Default frequency 12:00:00

Default auto-close enabled yes

21.2 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

21.3 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](#)

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

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

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 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

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

21.4 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 will start to fail if 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 will start to fail if 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

21.5 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 will make more room in the cache by freeing the least recently used queries from the cache, and then inserting the new queries. If this is happening often then you should increase the size of the cache to avoid this constant "swapping".

Default frequency 00:05:00

Default auto-close enabled no

Table Cache Not Optimal

MySQL is multi-threaded, so there may be many clients issuing queries for a given table simultaneously. To minimize the problem with multiple client threads having different states on the same table, the table is opened independently by each concurrent thread.

The table cache is used to cache file descriptors for open tables and there is a single cache shared by all clients. Increasing the size of the table cache allows mysqld to keep more tables open simultaneously by reducing the number of file open and close operations that must be done. If the value of [Open_tables](#) is approaching the value of [table_cache](#), this may indicate performance problems.

Default frequency 00:05:00

Default auto-close enabled no

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

21.6 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](#)

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.

Wrong Version Agent Tracker

Tracks wrong version agents that try to connect to this service manager.

21.7 Operating System Advisors

The **CPU Utilization** and **Filesystem Free Space** Advisors are described in [Chapter 22, GUI-Based Advisor Reference](#). The **Network Traffic Graphs** Advisor is used for graphing purposes, only, and has no configurable parameters other than the schedule.

RAM Usage Excessive

The **RAM Usage Excessive** Advisor monitors the amount of free RAM, in megabytes, on the monitored host.

This Advisor enables you to define thresholds, in megabytes of free RAM, for Notice, Warning, Critical, and Emergency.

21.8 Performance Advisors

This section describes the Performance Advisors.

- [Binary Log Usage Exceeding Disk Cache Memory Limits](#)
- [Database File I/O Global Summary](#)
- [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

Database File I/O Global Summary

Exposes the current summary of file I/O by wait type globally via the [sys.x](#) [\\$io_global_by_wait_by_latency](#) view.

This advisor has no configurable thresholds and is used to populate the graphs and tables of the **Database File I/O** report. The report continues to display historical data if the Advisor is disabled, but does not display any new data.

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 will block all SELECTs until the INSERT is completed. However, MyISAM can be configured to allow INSERT and SELECT statements to run concurrently in certain situations.

- If `concurrent_insert is set to 1` (the default, 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 `SELECT`s, thus effectively preventing `INSERT`s 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 that long-running statements will block a number of 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

21.9 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

21.10 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), you will trigger database errors and your application may not behave properly.

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 incoming data that is invalid will not be rejected by the server, but instead will be changed to conform to the target column's default datatype.



Note

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

21.11 Security Advisors

This section describes the Security Advisors.



Note

MySQL Enterprise Firewall and MySQL Enterprise Audit Plugin advisors are described in [MySQL Enterprise Firewall](#) and [MySQL Enterprise Audit Plugin](#).

- [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 Old Insecure Password Hash

Prior to MySQL 4.1, password hashes computed by the PASSWORD() function were 16 bytes long. As of MySQL 4.1 (and later), PASSWORD() was modified to produce a longer 41-byte hash value to provide enhanced security.

Default frequency 06:00:00

Default auto-close enabled no

Account Has Strong MySQL Privileges

Certain account privileges can be dangerous and should only be granted to trusted users when necessary. For example, the FILE privilege allows a user to read and write files on the database server (which includes sensitive operating system files), the PROCESS privilege allows currently executing statements to be monitored, and the SHUTDOWN privilege allows a user to shut down the server. In addition, the GRANT privilege allows a user to grant privileges to others.

Default frequency 00:05:00

Default auto-close enabled no

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 does not get loaded with server start, this will block access to the database for those users.

Default frequency 06:00:00

Default auto-close enabled yes

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 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 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 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 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

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

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

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 will be. 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

Server Contains Default "test" Database

By default, MySQL comes with a database named `test` that anyone can access. This database is intended only for testing and should be removed before moving into a production environment. Because the default `test` database can be accessed by any user and has permissive privileges, it should be dropped immediately as part of the installation process.

Default frequency 00:05:00

Default auto-close enabled no

Server Has Accounts Without A Password

Accounts without passwords are particularly dangerous because an attacker needs to guess only a username. Assigning passwords to all accounts helps prevent unauthorized users from accessing the system.

Default frequency 00:05:00

Default auto-close enabled yes

Server Has Anonymous Accounts

Anonymous MySQL accounts allow clients to connect to the server without specifying a username. Since anonymous accounts are well known in MySQL, removing them helps prevent unauthorized users from accessing the system.

Default frequency 00:05:00

Default auto-close enabled yes

Server 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, if this external authentication were to fail (such as the LDAP server losing power), then all administrator access to the MySQL Server will be 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

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 will regain privileges if a database with the same name is created in the future, which may not be the intended result.

Default frequency 00:05:00

Default auto-close enabled no

User Has Rights To Table That Does Not Exist

When a table is dropped, user privileges on the table are not automatically dropped. This has security implications as that user will regain privileges if a table with the same name in the same database is created in the future, which may not be the intended result.

Default frequency 00:05:00

Default auto-close enabled no

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 22 GUI-Based Advisor Reference

Table of Contents

22.1 Agent Health Advisor	207
22.2 MySQL Enterprise Backup Health Advisor	210
22.3 MySQL Process Discovery Advisor	211
22.4 Duplicate MySQL Server UUID	212
22.5 HTTP Server KeyStore's Certificate About to Expire	212
22.6 sys Schema Install Advisor	212
22.7 CPU Utilization Advisor	213
22.8 Filesystem Free Space Advisor	214
22.9 MySQL Process	216
22.10 Query Analysis Advisors	216
22.11 Security Advisors	217

This chapter describes the MySQL Enterprise Monitor GUI-based Advisors. That is, the advisors which are configured using a dialog rather than an expression.

22.1 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 22.1 Agent Health - General

The screenshot shows the 'General' tab of the 'Agent Health' configuration window. It contains the following settings:

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

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

- **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 22.2 Agent Health - General

The screenshot shows the 'Agent Health - General' configuration window. The 'Communication' tab is selected. It contains two main sections: 'Agent Latency Thresholds (minutes)' and 'HTTP Error Thresholds (% of total requests)'. In the first section, 'Warning Threshold' is checked with a value of 1 minute, and 'Critical Threshold' is checked with a value of 10 minutes. In the second section, 'Notice Threshold' is checked with a value of 10, 'Warning Threshold' is checked with a value of 20, and 'Critical Threshold' is checked with a value of 30. The 'Emergency Threshold' options are unchecked in both sections. 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 22.3 Agent Health - Backlog

The screenshot shows the 'Backlog' configuration window. It has three tabs: 'General', 'Communication', and 'Backlog'. The 'Backlog' tab is selected. Inside, there are two main sections. The first is 'Backlog Memory Usage Thresholds (% of max allowed)' with a help icon. It contains four checkboxes: 'Notice Threshold' (unchecked), 'Warning Threshold' (checked), 'Critical Threshold' (unchecked), and 'Emergency Threshold' (unchecked). Below these is a slider set to 80. The second section is 'Backlog Disk Usage Thresholds (% of max allowed)' with a help icon. It also has four checkboxes: 'Notice Threshold' (unchecked), 'Warning Threshold' (checked), 'Critical Threshold' (unchecked), and 'Emergency Threshold' (unchecked). Below these is another slider set to 80. At the bottom of the window 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.

22.2 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

22.3 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 22.1 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. • 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. 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 will be used in favor of a SUPER user,

Name	Description
	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. • 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

22.4 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, a Critical event is generated.

Default auto-close enabled yes

22.5 HTTP Server KeyStore's Certificate About to Expire

Alerts if the certificate expiration date falls within one of the defined thresholds. The thresholds are defined in numbers of days.

22.6 sys Schema Install Advisor

Alerts if `sys` schema is not installed on a monitored MySQL instance. This advisor also enables you to automatically install the `sys` schema. To install `sys` schema automatically on monitored instances, set **Install By Default** to **Yes**.

**Note**

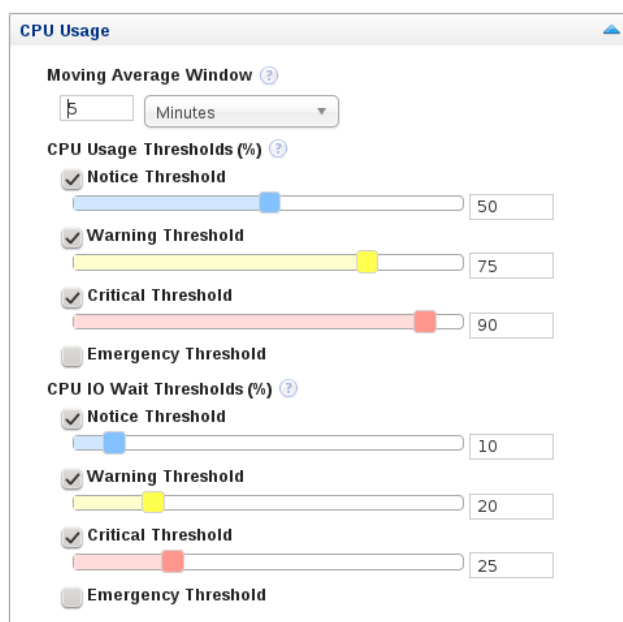
`sys` schema is supported on MySQL 5.6 and higher.

22.7 CPU Utilization Advisor

Monitors and graphs CPU usage on the monitored server or groups of servers.

CPU Usage

Figure 22.4 CPU Usage



- **Moving Average Window:** duration of the moving average window.
- **CPU Usage Thresholds:** configure the thresholds for percentage of total CPU usage.
- **CPI I/O Wait Thresholds:** configure the thresholds for CPU I/O 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 22.5 CPU Outliers

CPU Usage

CPU Outliers

Enable CPU Outlier Detection ?
No

Minimum Server Count For Outlier Detection ?
10

Small Group Notification ?
Yes

Outlier Percentile ?
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.

22.8 Filesystem Free Space Advisor

Monitors and graphs the filesystem disk space usage.

Default auto-close enabled yes

General

Figure 22.6 Filesystem - General

General

Analyze Filesystem Types ?

Local Disk x Network Mount x RAM Disk x Swap x

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

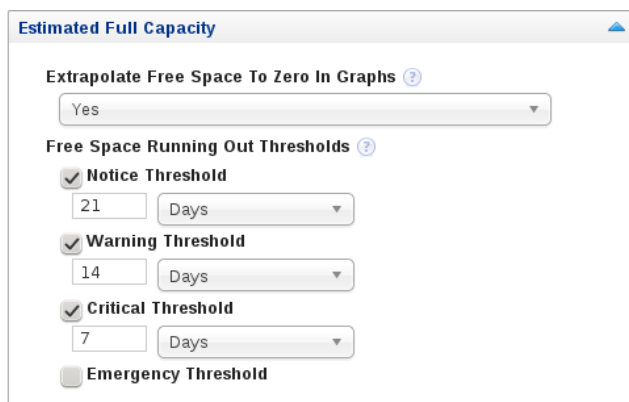
- **Local Disk:** enables monitoring of the local hard disks.
- **Network Mount:** enables monitoring of mounted network filesystems on the monitored server.
- **RAM Disk:** enables monitoring of RAM disks configured on the server.
- **CDROM:** enables monitoring of CD-ROM drives on the server
- **Swap:** 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 22.7 Filesystem - Estimated Full Capacity



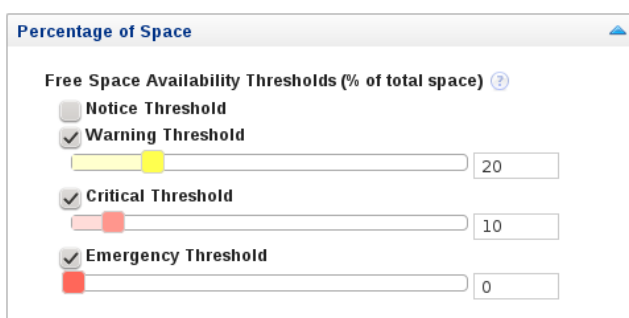
The screenshot shows a configuration window titled "Estimated Full Capacity". It contains two main sections. The first section, "Extrapolate Free Space To Zero In Graphs", has a dropdown menu set to "Yes". The second section, "Free Space Running Out Thresholds", contains four threshold settings, each with a checkbox and a value in days: "Notice Threshold" (checked, 21 days), "Warning Threshold" (checked, 14 days), "Critical Threshold" (checked, 7 days), and "Emergency Threshold" (unchecked, no value).

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 22.8 Filesystem - Percentage of Space



The screenshot shows a configuration window titled "Percentage of Space". It contains a section "Free Space Availability Thresholds (% of total space)" with four threshold settings, each with a checkbox and a percentage value: "Notice Threshold" (unchecked, no value), "Warning Threshold" (checked, 20%), "Critical Threshold" (checked, 10%), and "Emergency Threshold" (checked, 0%).

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 22.9 Filesystem - Percentage Used in Time Range

Monitors the percentage of disk space consumed per unit of time.

22.9 MySQL Process

The MySQL Process Advisor monitors and graphs MySQL CPU and memory utilization.

It is possible to edit the Moving Average Window size and change the schedule of this Advisor.

22.10 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 increases 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.

22.11 Security Advisors

This section describes the following **Security** advisors:

- [MySQL Enterprise Audit Plugin](#)
- [MySQL Enterprise Firewall](#)

MySQL Enterprise Audit Plugin

**Note**

For more information on the MySQL Enterprise Audit Plugin, see [MySQL Enterprise Audit](#).

This advisor enables you to configure event generation for the audit log plugin. This advisor has the following parameters:

- **Events Lost Threshold:** generates events for audit events which are lost due to setting the server's `audit_log_strategy` to `PERFORMANCE`. Enter a number of lost messages per threshold.
- **Write Wait Percent Thresholds:** generates events for the number of audit log write waits. The percentage is calculated as write waits versus writes.
- **Events Filtered Threshold:** generates events for the number of audit events which are filtered out by the audit log configuration
- **Detect Filtering Configurations:** if set to Yes, generates events for any configuration which filters audit log events. If set to No, such configurations are ignored.

MySQL Enterprise Firewall

**Note**

For more information on the MySQL Enterprise Firewall, see [MySQL Enterprise Firewall](#).

This advisor enables you to configure event generation for the MySQL Enterprise Firewall. This advisor has the following parameters:

- **Firewall Enabled Threshold:** generates events if the firewall is installed, but not enabled. To change the level of the alert, move the value 0 (representing "OFF") to the required threshold.
- **Access Denied Threshold:** generates events for the number of times statements were denied by the firewall. Enter the number of denials in the required thresholds.
- **Access Suspicious Threshold:** generates events for the number of times statements were deemed suspicious by the firewall.

Chapter 23 Access Control

Table of Contents

23.1 Users and Roles	219
23.2 Permissions	220
23.3 Monitored Assets Permissions	220
23.3.1 Server Group	221
23.3.2 MySQL Instances	221
23.4 Monitoring Services	223
23.5 MySQL Enterprise Monitor	223
23.6 Default Users and Roles	225
23.7 Creating Users and Roles	226

This chapter describes how to manage access to your MySQL Enterprise Monitor installation.

23.1 Users and Roles

MySQL Enterprise Monitor Access Control enables you to manage the following:

- Asset visibility: the rights to access data collected from hosts or MySQL instances. Access can be strictly limited to specific groups of monitored assets.
- Application administration: the rights to view or change the MySQL Enterprise Monitor configuration.
- Specific data access: the rights to view specific types of potentially sensitive data.
- Role reuse: rather than define permissions per user, permission sets are defined in Roles and multiple users can be assigned to each Role.

The MySQL Enterprise Monitor access control system is based on Users and Roles. Users have no rights assigned to them directly. All rights are defined on Roles. Users are assigned to Roles and inherit the rights defined on those Roles.

Roles

Roles are collections of permissions to which users are assigned. Roles define what the user is permitted to see and do in the application. Users can be assigned to multiple roles.

If users are assigned to multiple roles, MySQL Enterprise Monitor always takes the highest permission defined on those roles for that user. For example, if the user is assigned to a role with the **Advisor Configuration** set to Read-Only, and another role with **Advisor Configuration** set to Administer, Administer is the permission used for that user.

Users

Users are simple definitions of username, password, and an optional authentication method, such as Active Directory or LDAP. Each user must be assigned to at least one Role.



Note

It is not possible to save a user without an assigned Role.

23.2 Permissions

This section describes the permissions available in MySQL Enterprise Monitor Roles.

Permission Scope

There are two distinct permissions scopes in MySQL Enterprise Monitor:

- **System-wide Permissions:** apply to all assets and groups defined on the system. System-wide roles grant access to all monitored assets.
- **Group-specific Permissions:** grant access to specific groups of monitored assets. Permissions defined against a specific group apply to that group only. This setting affects everything the user sees. For example, Events are displayed for members of the group, only, and the status summary bar only displays information on the members of the group, and so on.



Important

It is not possible to assign permissions to the **All** group.

If you log in to the application as a group-specific user, the Asset Selector displays the group to which you are assigned, and the **All** group, which contains only those assets to which you have access.

Permission Groupings

Permissions are grouped in the following way:

- **Core Monitored Assets:** grant or deny access to the monitored assets and collected data.
- **MEM/Service Manager:** grant or deny access to the application and its settings.

Permission Types

The following grant types:

- **None:** no access to the functional area.
- **Read-Only:** read-only access to the functional area. The user can view, but not edit.
- **Administer:** complete access to the functional area. The user can view and edit.

ACL-related Error Messages

- If you have insufficient permissions to perform an action, the following message is displayed:

An Error Occurred. Access denied. You do not have sufficient permissions to perform the requested operation. (U0403)

23.3 Monitored Assets Permissions

The Core Monitored Assets permissions define access to the monitored assets, groups, and Query Analyzer data. The Monitoring Services permissions are dependent on these permissions.

Figure 23.1 Core Monitored Assets

Permission	☒ None	☐ Read-Only	☐ Administer
Server Group	☒	☐	☐
MySQL Instances	☒	☐	☐
Query Analysis Aggregate Data	☒	☐	☐
Query Analysis Example and Explain Data	☒	☐	☐

Defines the permissions for the monitored assets, groups, and visibility of the collected data in the Query Analyzer.



Important

Server Group and **MySQL Instances** are linked. If one is set to **Read-Only**, the other is set also. Similarly, if one is set to **Administer**, the other is set also. **MySQL Instances** requires **Server Group** be set to a value other than **None**. If **Server Group** is set to **None**, **MySQL Instances** is set to **None** also.

23.3.1 Server Group

Grants access to the monitored assets and groups. This permission must be used with the **MySQL Instances** permission.

- **None**: no access to any monitored asset. As a result, no information is displayed.
- **Read-Only**: Can view Groups of assets. This permission, or higher, is required for all other permissions which use Groups. Permissions such as **Event Handling** and **Server Group Creation** require access to the defined Groups. If the role requires access to those functional areas, this permission must be set.

Selecting Read-Only automatically selects Server Group Read-Only also.

- **Administer**: Can edit group information and delete groups of assets, but cannot create groups. Creating a group requires the **Server Group Creation** permission.

23.3.2 MySQL Instances

Grants access to the monitored instances. This permission must be used with the **Server Group** permission. If **Server Group** is set to Read-Only, or higher, it is impossible to set **MySQL Instances** to None. That is, if **Server Group** is set to Read-Only, or higher, **MySQL Instances** must be set to Read-Only at least.

- **MySQL Instances**: grants access to the data collected on the monitored MySQL Instances. Possible values are:
 - **None**: No access to MySQL Instances or the data collected on them.
 - **Read-Only**: access to the MySQL instances, but no rights to create, modify, or delete connections to those instances.
 - **Administer**: access to the MySQL instances, and can create, modify, and delete connections to those servers.

Administer is also required to access the bad connections, unreachable agents, and unmonitored instance lists on the **MySQL Instances** dashboard.

Administer is also required by the **Database File I/O**, which requires the [sys](#) schema. To install SYS schema from the MySQL Enterprise Monitor User Interface, the user must be assigned to a role with the **Administer** permission.



Warning

It is not possible to add, or start monitoring, a new instance without setting the **MySQL Enterprise Monitor** permission to Administer.

Query Analysis Permissions

The Query Analysis permissions define access to the Query Analysis page.

- **Query Analysis Aggregate Data:** access the data collected for the Query Analyzer. This permission also defines access to events which contain Query Analyzer data. Possible values are:
 - **None:** No access to the aggregated data collected for the Query Analyzer. If this permission is set, the user can open the Query Analyzer page, but the page does not load any aggregated data. This also affects the Query Analyzer graphs.

Events containing query analysis data are not displayed. Currently, this is limited to events generated by the **SQL Statement Generates Warnings or Errors** and **Average Statement Execution Time** advisors.

- **Read-Only:** Aggregated data is presented to the user, and the Query Analyzer page is populated.
- **Administer:** grants the right to close events containing Query Analysis aggregated data.
- **Query Analysis Example and Explain Data:** access the data for example and explain plans in the Query Analyzer. This permission depends on the **Query Analysis Aggregate Data** permission. This permission also defines access to events which contain EXAMPLE and EXPLAIN data. Possible values are:
 - **None:** no access is granted to the Query Analyzer EXAMPLE and EXPLAIN data.
 - **Read-Only:** EXAMPLE and EXPLAIN data is accessible. If **Query Analysis Aggregate Data** is not set to Read-Only, EXAMPLE and EXPLAIN data cannot be accessed.
 - **Administer:** grants the right to close events containing Query Analysis EXAMPLE and EXPLAIN data.



Note

The Query Analyzer permissions depend on the MySQL Instances permission. If MySQL Instances is set to Read-Only, both Query Analyzer permissions are also set to Read-Only. It is possible to set MySQL Instances to Read-Only, or higher, and manually set both Query Analyzer permissions to None, if required.

Monitored Asset Permission Dependencies

Each of the Monitored Asset permissions is dependent on the others. For a new role, all permissions default to None. Setting Server Group to Read-Only automatically sets all other Monitored Asset permissions to Read-Only. Similarly, if you set Server Group to Administer, MySQL Instances is also set

to Administer. It is not possible to set MySQL Instances to None if Server Group is set to Read-Only or higher.

23.4 Monitoring Services

Figure 23.2 Monitoring Services Permissions

Permission	☒ None	☐ Read-Only	☐ Administer
Agent Services access ?	☒	☐	(n/a)
Web Application Login ?	☒	☐	(n/a)
MySQL Enterprise Monitor ?	☒	☐	☐
Advisor Configuration ?	☒	☐	☐
Event Blackout ?	☒	(n/a)	☐
Event Handling ?	☒	☐	☐
Server Group Creation ?	☒	(n/a)	☐
Settings ?	☒	☐	☐
Users and Roles ?	☒	☐	☐

Agent Services Access

This permission is for use by Agent Roles only. Possible values are:

- **None:** No Agent access.
- **Read-Only:** Agent has access to the MySQL Enterprise Service Manager



Important

If you are defining an agent role, you must set all other permissions to None. The agent does not require them.

Web Application Login

The **Web Application Login** permission grants access to the application interface.

- **None:** no access to the user interface.
- **Read-Only:** access to the user interface.

23.5 MySQL Enterprise Monitor

The **MySQL Enterprise Monitor** permission grants access to the various configuration settings of the MEM interface. Possible values are:

- **None:** no access to the configuration settings.
- **Read-Only:** configuration settings are visible, but cannot be edited.

Authentication-level settings, such as the **External Authentication** and **HTTP Proxy Settings** on the **Settings** page are not visible if this is set to Read-Only.

- **Administer:** configuration settings are visible and editable.

Setting any of these values automatically sets the same value for all nested permissions.

**Note**

The following permissions require **Web Application Login** and **MySQL Enterprise Monitor** set to Read-Only, or higher.

Advisor Configuration

Advisor Configuration defines access to the Advisor page and its settings.

**Note**

Advisors do not run as the user who created or enabled them, but as the system role. This is done to avoid problems such as user deletion, replication topology visibility (advisors collect on complete topology, but user may only see part of that topology). As such, the Advisors cannot be set on a group-specific level; they can only be set at a global level.

- **None:** no access to the Advisors. If the user attempts to load the Advisors page, an Access Denied error is displayed.
- **Read-Only:** read-only access to the Advisors. The user can view the Advisors, but cannot save changes.
- **Administer:** the user has complete access to the Advisors.

**Important**

Overriding an Advisor at the top-level, not on an individual asset, overrides that Advisor globally, for all users, regardless of their roles. If an Advisor's schedule is changed, or disabled, at the top-level, it affects all users of MySQL Enterprise Service Manager regardless of their group setup.

Event Blackout

Event Blackout: Possible values are:

- **None:** no access to **Event Handler Blackout** menu on **MySQL Instances** dashboard.
- **Administer:** **Event Handler Blackout** menu is displayed and can be selected.

Event Handling

The **Event Handling** permission grants access to the **Event Handling** page and menu item. Possible values are:

- **None:** no access to Event Handling. The **Event Handling** menu item is not displayed on the **Settings** menu.
- **Read-Only:** read-only access to **Event Handling**. The **Event Handling** page is accessible, but it is not possible to create, delete, or edit event handlers.
- **Administer:** full access to **Event Handling** page. Users associated with this role can create, edit, suspend, and delete Event Handlers.

**Note**

If the user does not also have **Server Group** set to at least Read-Only, they are unable to add groups to an Event Handler.

New Group Creation

The **New Group Creation** permission enables creation of groups. Possible values are:

- **None**: no access to server group creation. If **Server Group** is set to Administer, assigned users can delete and modify existing groups, but cannot create new groups.
- **Administer**: full access to server groups. If **Server Group** is set to Administer, the assigned user can create, delete, and edit server groups. If **Server Group** is set to Read-Only, the assigned user can create new groups, but cannot modify existing groups.



Note

This permission depends on the Server Group permission. If Server Group is set to None, the user associated with this role cannot access groups and, as a result, cannot create or edit groups, even if New Group Creation is set to Administer.

Settings

The **Settings** permission grants access to the **Settings** menu item and **Settings** page. Possible values are:

- **None**: the Settings menu item is not displayed.
- **Read-Only**: read-only access to the Settings. Assigned users can open the Settings page, but cannot change any settings.
- **Administer**: full access to the Settings. Assigned users can open the Settings page and edit the values.



Important

Settings this permission to Administer does not grant access to the External Authentication section of the **Settings** page.

Users and Roles

Users and Roles: Possible values are:

- **None**: no access to the User or Roles pages.
- **Read-Only**: read-only access to the Users and Roles pages. Assigned users can view, but not edit.
- **Administer**: full access to the Users and Roles pages. Assigned users can view and edit both Users and Roles.

23.6 Default Users and Roles

The default roles enable migration of defined roles from earlier versions. It is not possible to edit the default roles.

Default Users

The following default users are created when MySQL Enterprise Service Manager is first installed and setup:

- **Agent user**: defines the username and password used by all agents to connect to MySQL Enterprise Service Manager. This user is automatically added to the Agent role. The username defined on the initial setup page is used.

- The Manager user: defines the username and password of the Manager user. This user is automatically added to the Manager role which has all rights granted. The username defined on the initial setup page is used.

Default Roles

This section describes the default roles.



Important

It is not possible to edit or delete the default roles. They are present to enable upgrades from earlier versions, only.

The following are the default roles and a brief explanation of how they map to user definitions from earlier versions:

- **agent**: the role used by the agent user. This role has only the **Agent Services access** permission defined because the agent does not need access to any MySQL Enterprise Service Manager functionality.
- **dba**: maps to the dba role from previous versions. Any user with dba defined in 3.0, is added to dba in 3.1.
- **Display Query Analyzer**: maps to **View Query Analyzer tab** in 3.0. Any user with **View Query Analyzer tab** defined in 3.0, is added to the **Display Query Analyzer** role in 3.1.
- **Display Query Analyzer Examples**: maps to **View actual (example) queries** in 3.0. Any user with **View actual (example) queries** defined in 3.0, is added to the **Display Query Analyzer Examples** role in 3.1.
- **manager**: maps to the manager role in previous versions.
- **readonly**: maps to the readonly role in previous versions.

Users are added to the default roles based on the rights assigned to them in the earlier version of MySQL Enterprise Monitor. For example, if a user is assigned to the dba role and has both View Query Analyzer tab and View actual (example) queries enabled, the user will be added to the following Roles in 3.1:

- dba
- Display Query Analyzer
- Display Query Analyzer Examples

23.7 Creating Users and Roles

This section describes how to create users and roles.

Creating a Role



Note

It is not possible to save a new user without an assigned role. It is recommended to create Roles before creating Users.

To create a role, do the following:

1. Select **Roles** from the Settings menu (gear icon). The **Roles** page is displayed.
2. On the **Roles** page, click **Create**. The **Create Role** page is displayed.
3. On the **Details** tab, enter a name in the **Role Name** field and add a description of the role.

If you are using an external authentication system, such as LDAP or Active Directory, enter the external role name in the **External Roles** field.
4. Click **Permissions** to open the **Permissions** tab.
5. If this role applies to a specific group only, select **Group-Specific Permission**, and select the required group from the drop-down list.
6. Define your permissions as required. For more information, see [Section 23.3, "Monitored Assets Permissions"](#) and [Section 23.4, "Monitoring Services"](#)
7. If users exist, you can add them to this Role using the **Assigned Users** tab.

To add a user, click on the user name in the **Available Users** field. The user is moved to the **Assigned Users** field.
8. Click **Save** to save your changes, or click **Cancel** to discard your changes.

Creating a User

This section describes how to create a user.

To create a user, do the following:

1. Select **Users** from the Settings menu (gear icon). The **Users** page is displayed.
2. Click **Create**. The **Create User** page is displayed.
3. Enter the following:
 - **User Login**: the username the user will use to login.
 - **Full Name**: the user's full name.
 - **Password**: the user's password.
 - **Confirm Password**: enter the user's password again.
 - **Authenticate this user using LDAP**: select only if you intend to use LDAP to authenticate this user.
4. It is not possible to save a user without assigning the user to a Role.

Select the **Assign Roles** tab.
5. Assign roles to the user by clicking the required role in the **Available Roles** field.
6. Save your changes.



Important

It is not possible to edit a user's role, if the user is authenticated by LDAP and their role is also provided by LDAP.

Chapter 24 Access Control - Best Practices

Table of Contents

24.1 Open Permission Sets	230
24.2 Strict Permission Set	232

This chapter describes some best practices for setting up your access control permissions. As each organisation has a different way of implementing their MySQL installations and monitoring, the scenarios described are general guidelines.

The following scenarios are described:

- Open: an organisation with one, or more, DBAs. All users can see, but have varying access to, all monitored assets.
- Strict: an organisation with several DBAs and developers, and many monitored assets, grouped according to the applications and users which use them. Some users within the organisation have access to all monitored assets, some have access only to a subset of those assets and cannot see any asset which falls outside their responsibilities. This scenario adopts a production vs. development pattern.

Typically, in this type of scenario, there is a strict separation between production and development. That is, those roles which have complete access on the development assets, have only limited access, or no access, to the production assets.

The roles involved in each scenario are as follows:

- Database Administrator (DBA): responsible for the proper operation of the MySQL instances. As such, they need access to the data collected on the monitored instances. In most scenarios, the DBA can access the majority of the MySQL Enterprise Monitor functionality, such as Advisors, Event Handlers, and Query Analysis.



Note

While there is a default DBA role included in your installation, it is recommended to create a separate DBA-type role for your installation. The default DBA role exists to facilitate migration from previous versions. Also, it is not possible to edit the default DBA role.

For the purposes of this chapter, the DBA role is taken by SeniorDBA and JuniorDBA.

- Group/User Administrator: responsible for user, role, and group management. This role defines who has access to MySQL Enterprise Service Manager and defines the grouping of the servers. Users in this role are typically high-level DBAs, IT administrators, or project managers. In large organisations, the Group Administrator role may also be responsible for managing Event Handlers, Event Blackouts, and Notification Groups. It is strongly recommended that a group administrator is assigned in all setups. The scope of the Group Administrator role's permissions can vary, depending on the size of the organization. In smaller organisations, members of this role are solely responsible for the addition of users, roles and groups. While, in larger organisations, they are also responsible for managing the event traffic via email/SMTP notifications, group management, and so on.

The GroupAdmin role is a lock-and-key role. It is defined in such a way that it cannot be used on its own. To add groups, users or roles, it must be used in combination with a role which grants the top-level

permissions, **Server Group** and **MEM Web Application**. That is, for a user to have permissions to edit users, roles and groups, they must be members of both the GroupAdmin role and another role which grants the dependent permissions.

The GroupAdmin role is recommended for all implementations except the most basic.

- Developers: responsible for the code deployed on the assets. As such, they need to see the impact of their code on the monitored assets. In a production environment, the developers have access to Events, Query Analysis, graph data, and so on.

24.1 Open Permission Sets

The Open implementation has no group-specific roles. This scenario has the following role types:

- Manager: responsible for all monitored assets, advisor configuration, group configuration, query analysis, event handling and communications. (Default role. Complete access.)
- DBA: responsible for monitored assets, query analysis, event investigation.

The following users are involved in this scenario:

- Manager: responsible for all monitored assets.
- DBA: responsible for monitoring MySQL instances, investigating issues and repairing those issues.

Manager Role

This section describes the Manager role definition for the Open implementation. Users in this role are power users. They are responsible for configuring everything. This role is permitted to perform the following actions:

- All possible actions .

Table 24.1 Manager Role Definition

Permission	Level
Server Group	Administer
MySQL Instances	Administer
Query Analysis Aggregate Data	Read-Only
Query Analysis Example and Explain Data	Read-Only
Web Application Login	Read-Only
MySQL Enterprise Monitor	Administer
Advisor Configuration	Administer
Event Blackout	Administer
Event Handling	Administer
New Group Creation	Administer
Settings	Administer
Users and Roles	Administer

The Manager users are responsible for configuring Advisor thresholds and defining the Event Handlers and Notification Groups. The Notification Groups contain the members of the standard DBA role, and the Senior DBA members.

This user has the permission to close Events, due to **MySQL Instances** being set to Administer.

DBA Role

This section describes the DBA role definition for the Open implementation. Users in this role are monitoring users. They are responsible for investigating events and resolving issues with the monitored MySQL instances. This role is permitted to perform the following actions:

- All tasks except User Management and editing MEM settings.

Table 24.2 DBA Role Definition

Permission	Level
Server Group	Administer
MySQL Instances	Administer
Query Analysis Aggregate Data	Read-Only
Query Analysis Example and Explain Data	Read-Only
Web Application Login	Read-Only
MySQL Enterprise Monitor	Read-Only
Advisor Configuration	Administer
Event Blackout	Administer
Event Handling	Administer
New Group Creation	Administer
Settings	None
Users and Roles	None



Note

It is possible, in this Open implementation, to add all DBA users to the default DBA role. However, for any size installation, it is recommended to have a well-defined hierarchy of users. Particularly for SMTP or SNMP notifications which can, if unmanaged, produce a very high volume of notification traffic. It is recommended to have a single group of senior users manage Advisor, Event Handler, and Notification Group configuration. All requests should go through those senior users.

Also, it is not possible to edit the default DBA role.

Role Membership

Users are assigned to roles in the following way:

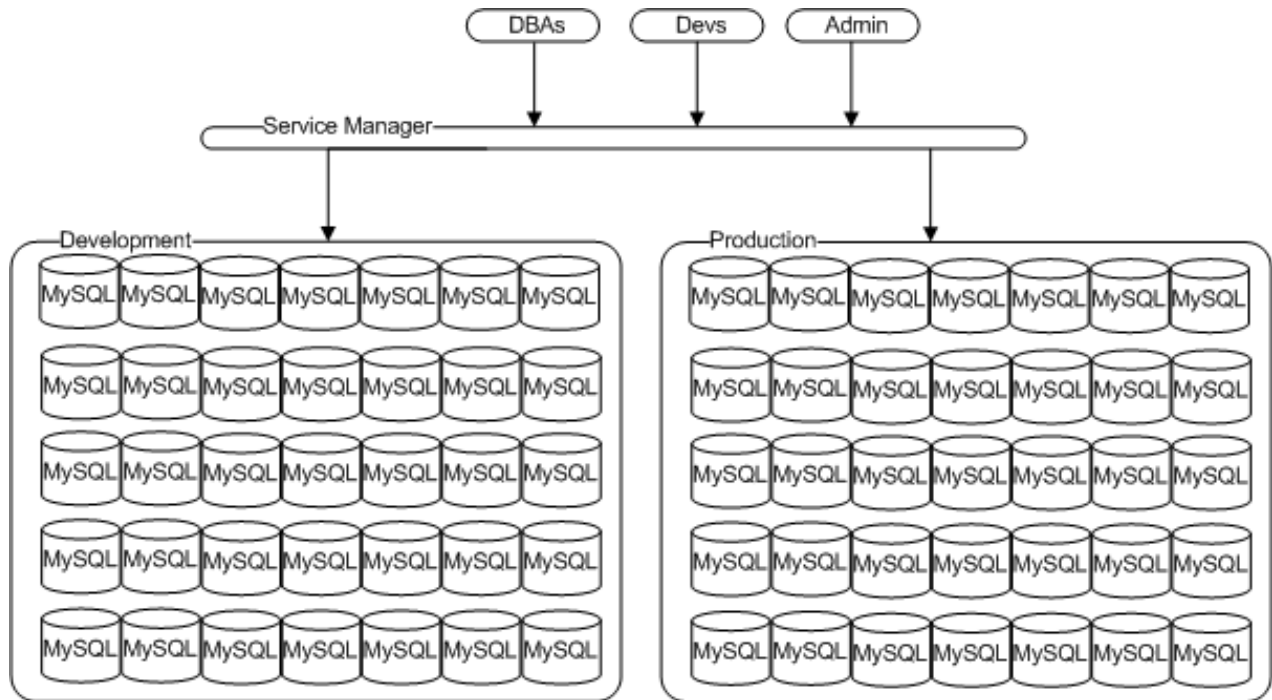
- Manager Role
 - Teamlead/Coordinator user
- DBA Role
 - DBAs

24.2 Strict Permission Set

The Strict scenario is a group-based implementation. Users are assigned to roles with varying access to the groups.

This scenario focuses on two groups, Development and Production. Development is the group of MySQL instances where the product is developed and tested. Production is the group of MySQL instances to which the finished product is deployed for customers to use.

Figure 24.1 Hybrid Permission Set Overview



Users, Roles and Groups

This implementation requires the following asset groups:

- Development: all assets used by the development and quality teams are grouped in the Development group.
- Production: all assets deployed for use by the customer are grouped in the Production group.



Note

When installing agents to monitor the assets, it is critically important to choose the correct group during the installation process. If the incorrect group, or no group, is chosen, the assets fall outside the scope of the Roles defined here and cannot be seen by any user except those in the SeniorDBA roles.

This implementation requires the following roles types:

- GroupAdmin: System-wide role. Members are responsible for user, role, and group management only. This role is limited in the sense that it does not have the **Server Group** or **MEM Web Application**

permission set to a usable value. To access the UI or create groups, the users assigned to this role must also be assigned to roles with usable Server Group permissions (Read-Only or Administer).

- **SeniorDBA:** System-wide role. Members have access to all monitored assets on both Production and Development groups. No group-specific permission sets.
- **JuniorDBA:** members have read-only access to the monitored assets in the Development group, only.
- **SeniorDev-Development:** members have limited access to monitored assets in Development group. Members of this role need permissions to view events, QuAn data, and create event handlers on the Development assets. Members of this role are responsible for inspecting the impact their code has on performance and existing functionality.
- **SeniorDev-Production:** Same members as SeniorDev-Development, but restricted rights on the monitored assets. Permissions to observe, only, no rights to create event handlers, set blackouts, or access the QuAn Explain or Example functionality. This role does not include any observation of customer data, but does allow its members to view events generated on the monitored assets.

If a member of this role requires an event handler or advisor threshold edit on the Production group, it must be requested from a member of the SeniorDBA role.

- **JuniorDev-Development:** members have access to the Development group, only. For the most part, their permissions are read-only. They are entitled to view events, QuAn data, and so on.

This implementation requires the following users:

- **DBA Teamlead:** manages the DBA team and has complete access to all monitored assets. This user is a member of the SeniorDBA and GroupAdmin roles. This combination of permissions gives them complete access to all monitored assets.
- **Senior DBAs:** responsible for the monitored assets. Has complete access to all monitored assets. No user management rights.
- **Junior DBAs:** responsible for investigating issues. Read-only rights on all Development assets. No access to Production assets.
- **Senior Developers:** responsible for deploying code to the Development group and reviewing impact on performance and functionality. No user management rights, event blackout rights, and so on. Permitted to view events on the Production group, but not to add event handlers, notification groups, and so on.
- **Junior Developers:** responsible for deploying code and viewing events on the Development group. No access to the Production group.

System-Wide Role Definitions

For each of these roles, select **System-Wide Permissions** in the **Core Monitored Assets** frame.

Table 24.3 System-Wide Role Definition

Permission	SeniorDBA	GroupAdmin
Server Group	Administer	None
MySQL Instances	Administer	None
Query Analysis Aggregate Data	Administer	None
Query Analysis Example and Explain Data	Administer	None

Permission	SeniorDBA	GroupAdmin
Web Application Login	Read-Only	None
MySQL Enterprise Monitor	Administer	None
Advisor Configuration	Administer	None
Event Blackout	Administer	None
Event Handling	Administer	None
New Group Creation	None	Administer
Settings	Administer	None
Users and Roles	None	Administer

The membership of these Roles is:

- SeniorDBA Role: DBA manager and Senior DBAs.
- GroupAdmin: DBA manager and at least one Senior DBA, for redundancy.

Development Group Roles

For each of these roles, select **Group-Specific Permissions** in the **Core Monitored Assets** frame, and select **Development** from the group drop-down list.

Table 24.4 Development Group Role Definition

Permission	SeniorDev	JuniorDev	JuniorDBA
Server Group	Administer	Read-Only	Read-Only
MySQL Instances	Read-Only	Read-Only	Read-Only
Query Analysis Aggregate Data	Read-Only	Read-Only	Read-Only
Query Analysis Example and Explain Data	Read-Only	Read-Only	Read-Only
Web Application Login	Read-Only	Read-Only	Read-Only
MySQL Enterprise Monitor	Read-Only	Read-Only	Read-Only
Advisor Configuration	Read-Only	Read-Only	Read-Only
Event Blackout	None	None	None
Event Handling	Read-Only	None	Read-Only
New Group Creation	None	None	None
Settings	None	None	None
Users and Roles	None	None	None



Note

Currently, **Advisor Configuration** and **Event Handling** are global permissions. Changes made at that level can affect all users of the MySQL Enterprise Monitor. As such, only a senior user, with System-Wide permissions, should be permitted to change these settings.

Production Group Roles

For this role, select **Group-Specific Permissions** in the **Core Monitored Assets** frame, and select **Production** from the group drop-down list.

Table 24.5 Production Group Role Definition

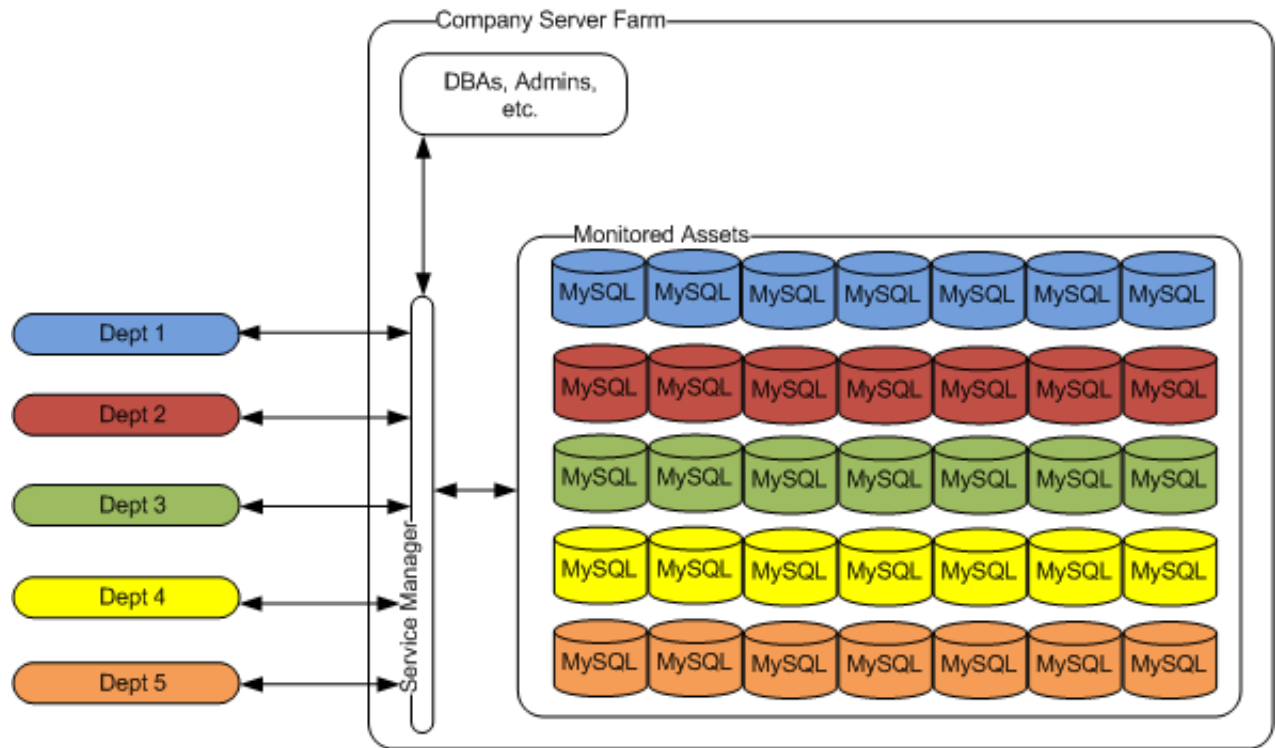
Permission	SeniorDev
Server Group	Read-Only
MySQL Instances	Read-Only
Query Analysis Aggregate Data	None
Query Analysis Example and Explain Data	None
Web Application Login	Read-Only
MySQL Enterprise Monitor	Read-Only
Advisor Configuration	Read-Only
Event Blackout	None
Event Handling	None
New Group Creation	None
Settings	None
Users and Roles	None

Distributed Departments

The Strict implementation is also useful for large companies with globally distributed teams, accessing central server farms.

This implementation involves the following:

- Company server farm with DBAs and individuals responsible for liaising with departments.
- Departments with their own DBAs, Developers, and so on. This implementation includes the following departments, each with an identical permissions set: BlueTeam, RedTeam, GreenTeam, YellowTeam, and OrangeTeam.
- Groups must be configured for each department. In this scenario, BlueGroup, RedGroup, GreenGroup, YellowGroup, and OrangeGroup. Where each group contains the assets dedicated to each department.

Figure 24.2 Strict Permission Set Grouped

Chapter 25 Global Settings

Table of Contents

25.1 Server Locale	237
25.2 Server Hostname	237
25.3 Customize MySQL Server Name	237
25.4 Data Purge Behavior	239
25.5 My Oracle Support Credentials	239
25.6 HTTP Proxy Settings	240
25.7 External Authentication	240

This chapter describes how to configure your MySQL Enterprise Service Manager installation.

25.1 Server Locale

This locale overrides the operating system locale for use in notifications. Select your locale from the list of options.

25.2 Server Hostname

This section describes how to define your server hostname.

Figure 25.1 Server Hostname

The screenshot shows a 'Server Hostname' configuration window. It contains three main sections: 'Hostname' with a text box containing 'my.mem.com' and a note that it overrides the OS hostname; 'Port Number' with a text box containing '18443' and a note that it overrides the Apache Tomcat port; and 'Login Display Name' with a text box containing 'MEMServer1' and a note that it overrides the OS hostname on the login page. A 'Save' button with a green checkmark is at the bottom left.

Table 25.1 Customize

Name	Description
Hostname	Defines the hostname used in all notifications. This value must be a valid hostname.
Port Number	Defines the port number used for notifications. Do not change this value from the default unless you have altered or redirected the default port number during installation. An invalid value results in invalid links in notification messages. Default value is 18443.
Login Display Name	Defines the hostname displayed on the login page.

25.3 Customize MySQL Server Name

This section describes how to configure the display names of your monitored MySQL hosts.

Figure 25.2 Customize MySQL Server Name

Table 25.2 Customize

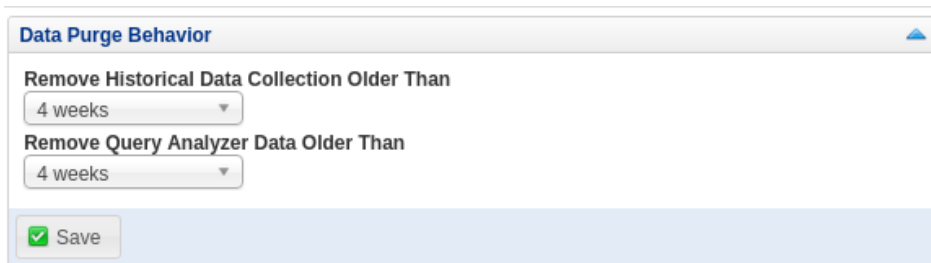
Name	Description
Show MySQL Server Names as:	Defines how hostnames are displayed. The following display settings are possible: • Reported by Agent (default): the MySQL server names are displayed exactly as reported by the monitoring agent. • Hostname only: only the host name is displayed, omitting suffixes such as companyname.com. • Transformed by substitution expression: enables the Substitution Expression field. This enables you to replace some, or all, of the hostname with custom values.
Substitution Expression	Enables you to substitute your hostnames with custom values. These substitutions can be simple substitutions, or more complex regular expressions. The substitution expression is a name-value pair, with the original value, or regular expression on the left, and the substitution value on the right. You can include multiple, comma-separated, substitutions. For example: <pre>dx521\example.com=Staging dx984[.]example[.]com=Production database-server-(.*?)\example.com=\$1 ^database-server-="", [.]example[.]com\$="", dx521="Staging DB", dx984="Production"</pre> More complex substitutions are possible by using a regular expression. For more information on the syntax used in these regular expressions, see Regular Expression Constructs.
Display Connection Endpoint Values	Defines how connection endpoint values are displayed. The following display settings are possible: • Always (default): endpoint values are always displayed. • Never: endpoint values are never displayed.

Name	Description
	• For non-default values: endpoint values are displayed only if they differ from 3306 and /tmp/mysql.sock.

25.4 Data Purge Behavior

Data Purge Behavior enables you to automatically remove old log files data from the repository according to a schedule. 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 25.3 Data Purge Behavior



Purging data permanently removes information from the repository. Events derived from that data are purged with the data.



Note

The purge functionality purges closed events and related data, only.

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.

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 [innodb_file_per_table](#) for the repository database and 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.

25.5 My Oracle Support Credentials

You can specify the credentials for logging into the My Oracle Support site. These must match the user name and password that you have registered with Oracle for access to the support site.

Figure 25.4 My Oracle Support Credentials

My Oracle Support Credentials

These credentials are used to retrieve and display your open Service Requests (for those with an active customer account only).

My Oracle Support Login (Email Address)

Password

Confirm Password

☒ Save

25.6 HTTP Proxy Settings

Enter your HTTP Proxy details, if you use a proxy to connect to the internet. If you use a proxy, and these values are not set, the What's New frame cannot update.

Figure 25.5 HTTP Proxy Settings

HTTP Proxy Settings

Proxy Host:port

Proxy Username

Proxy Password

Confirm Password

☒ Save

25.7 External Authentication

Table 25.3 External Authentication

Name	Description
Disabled	No external authentication system is used. All user authentication is performed in MySQL Enterprise Monitor.
LDAP Authentication	Enables the LDAP configuration. Populate the fields as required by your LDAP installation.
Active Directory Authentication	Enables the Active Directory configuration. Populate the fields as required by your Active Directory installation.
Is Authoritative	To make the selected authentication system the authoritative authentication mechanism, check Is Authoritative .

Name	Description
	 Important If you select this option, and the LDAP service is misconfigured, you can lock yourself out of MySQL Enterprise Monitor entirely.

External Authentication

Enables you to configure external authentication using LDAP or Active Directory.

Figure 25.6 External Authentication Settings: LDAP

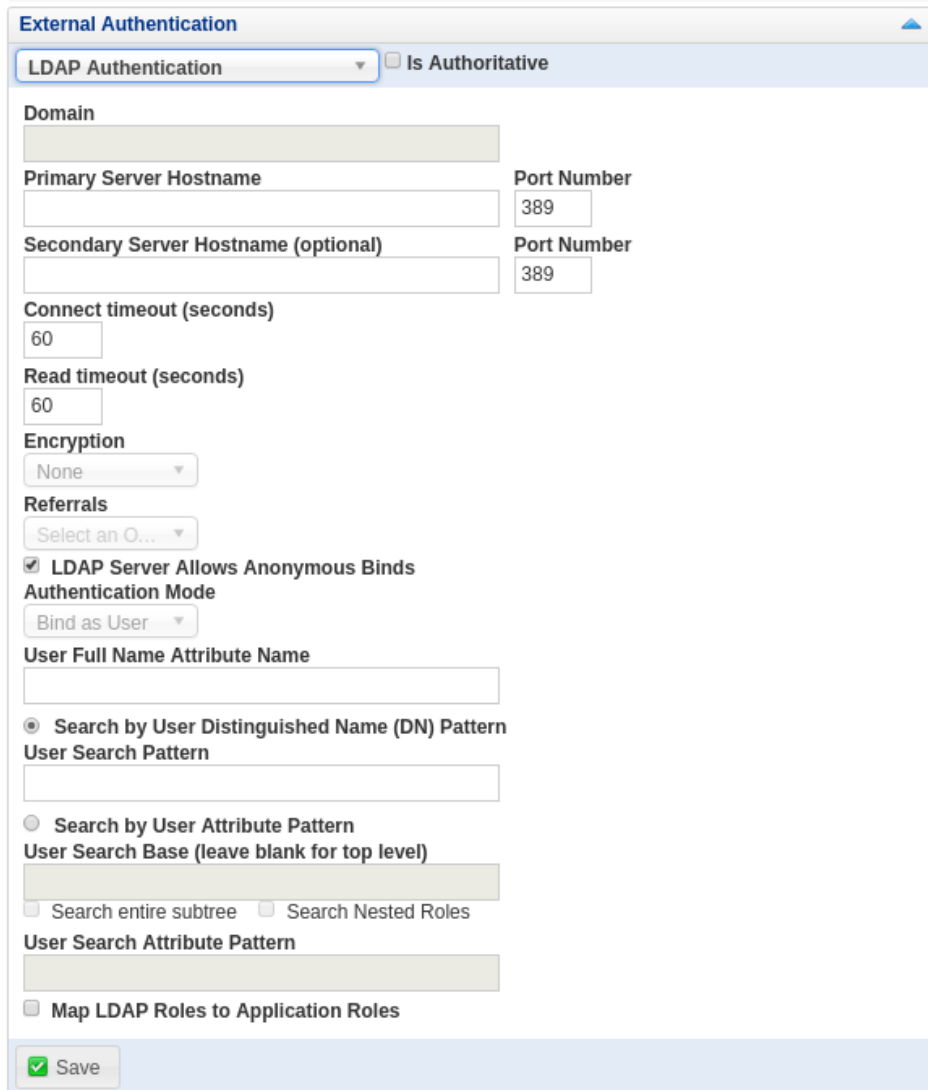


Table 25.4 LDAP Authentication

Name	Description
Primary Server Hostname and Port Number	Hostname or IP address of the primary LDAP directory server, and the Port number of the primary LDAP server. You must

Name	Description
	change this option to the port used for SSL connections if you have enabled encryption.
Secondary Server Hostname and Port Number	Hostname or IP address of the secondary LDAP directory server. Port number of the secondary LDAP server. You must change this option to the port used for SSL connections if you have enabled encryption.
Connect Timeout (seconds)	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.
Read Timeout (seconds)	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.
Encryption	Encryption type required for communication with the LDAP server(s). Supported options are None , StartTLS , and SSL .
Referrals	Authentication follows the referrals provided by the server. The default is to use whatever the LDAP directory server is configured to do.
External Authentication Server Allows Anonymous Binds	Optionally allow Anonymous binds. When unchecked, MySQL Enterprise Monitor provides for a pre-auth bind user to lookup account records. For 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, it cannot fetch credentials.
Authentication Mode	The authentication mode to use. • Bind as User: binds to the LDAP directory using the credentials given to login to MySQL Enterprise Service Manager • Comparison: requires an LDAP login/password that can see the configured password attribute to make a comparison with the given credentials.
User Full Name Attribute Name	Define the user fullname attribute. This enables the system to return the fullname of the user.
Search by User Distinguished Name (DN) Pattern	In the User Search Pattern field, define the pattern specifying the LDAP search filter to use after substitution of the username, where {0} defines where the username should be substituted for the DN.
Search by User Attribute Pattern	In the User Search Base (leave blank for top level) field, define the value to use as the base of the subtree containing users. If not specified, the search base is the top-level context. To search the entire subtree, starting at the User Search Base Entry, enable Search entire subtree. If disabled, a single-level search is performed, including only the top level. To include nested roles in the search, enable Search Nested Roles.

Name	Description
User Search Attribute Pattern	The attribute pattern to use in user searches.
Map External Roles to Application Roles	Specifies whether the roles defined in LDAP should 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, additional fields are displayed which enable you to configure how roles are found in the LDAP server.

Active Directory Authentication

Enables you to configure Active Directory authentication.

Table 25.5 Active Directory Authentication

Name	Description
Domain	The Active Directory Domain.
Primary Server Hostname and Port Number	Hostname of the Active Directory server to use.
Secondary Server Hostname and Port Number (optional)	Secondary Active Directory hostname. This is optional.
Map LDAP Roles to Application Roles	Whether the roles defined in Active Directory can be mapped to those defined in MySQL Enterprise Monitor.

Chapter 26 Customizing MySQL Enterprise Monitor

Table of Contents

26.1 Creating Advisors and Rules	245
26.1.1 Creating Advisors	245
26.1.2 Overview of Graph Creation	246
26.1.3 Overview of Advisor Creation	247
26.1.4 Variables	248
26.1.5 Thresholds	248
26.1.6 Using Strings	249
26.1.7 Wiki Format	250
26.1.8 Creating a New Advisor: An Example	250
26.1.9 Creating a New Graph: An Example	251
26.2 Custom Data Collection	252
26.2.1 Custom.xml	253
26.2.2 Queries	253
26.2.3 Data Collection Attributes	255
26.3 Event Notification Blackout Periods	256
26.3.1 Scripting Blackouts	257

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.

26.1 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.

26.1.1 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.

26.1.2 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 26.1.9, “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.

26.1.3 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.

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.

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.

26.1.4 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.”

26.1.5 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) < THRESHOLD)
```

The first part of the expression, `(%Uptime% > 10800)`, delays evaluating this expression until the system has been running for 10800 seconds (3 hours). When a server starts up, it might take a while to reach a state that is representative of normal operations. For example, the **InnoDB** buffer pool, **MyISAM** key cache, and the SQL query cache might require some time to fill up with application data, after which the cached data boosts performance.

In addition, if some part of the system is not heavily used, an alert might be triggered based on limited data. For example, if your application does not use the MyISAM storage engine, the “MyISAM Key Cache Has Sub-Optimal Hit Rate” rule could be triggered based on very limited use of other MyISAM tables such as the `mysql.user` table. For this reason, this advisor has a second part: `(%Key_read_requests% > 10000)`. The rule is not evaluated unless there is plenty of activity associated with the key cache.

26.1.6 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.

26.1.7 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 26.1 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 <code>'\'</code> 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>
```

26.1.8 Creating a New Advisor: An Example

This section documents the steps to create an Advisor.

To create an Advisor, select the **Create Advisor** button from the **Advisors** page. The new advisor page is displayed.

This example creates an Advisor that checks if connections have been killed using the **KILL** statement and generates an event.

Create your custom rule by following these steps:

1. Using the **Advisor Name** text box, give the Advisor an appropriate name, such as "Connections killed".
2. From the **Advisor Category** drop down list box, choose an Advisor category for your Advisor.
3. Define the variable for your expression in the **Variable Assignment** frame.
 - In the **Variable** text box, enter `%connections_killed%`, the variable used in the **Expression** text box.
 - In the **Data Item** drop-down list, select the `mysql:status:Com_kill` entry.
 - 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"](#).

26.1.9 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 26.1.2, "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.

26.2 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.

26.2.1 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 26.2 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 used to check the version of the MySQL server. See Section 26.2.2.1, "Precondition Queries" for more information.
<code>query</code>	The main query. For more information, see Section 26.2.2.2, "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 26.2.3, "Data Collection Attributes" for more information.

The values in the namespace and classname elements are used as the first two elements of the name.

26.2.2 Queries

This section describes the precondition and main queries used to create custom data collections.

26.2.2.1 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.

26.2.2.2 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 26.2.3.2, “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.

26.2.2.3 Wiki Formatting in Queries

It is possible to format the query result with wiki markup. This enables you to display information from the query directly in the event generated by the advisor.

The following example is taken from the data collection used by the **Server Has Accounts Without A Password** advisor:

```
<query>
<![CDATA[SELECT GROUP_CONCAT('\n* ', '\', user, '\', host, '\', ORDER BY user, host)
as user FROM mysql.user WHERE password=' /*!50507 AND (plugin = '' OR plugin IS NULL
OR plugin = 'mysql_native_password') OR (plugin = 'sha256_password'
AND authentication_string = '')*]]>
</query>
```

The wiki markup formats the user and host into information readily displayed in the Events page of MySQL Enterprise Monitor User Interface. This example lists the user name and host for all accounts without a defined password.

See [Section 26.1.7, “Wiki Format”](#) for more information on the supported wiki markup.

26.2.3 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 26.3 Attribute Elements

Name	Description
name	The name of the attribute defined in the AS clause of the query.
counter	Whether the attribute is a counter type. true: the attribute is a counter type. false: 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.

26.2.3.1 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>
```

26.2.3.2 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>
```

26.3 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.

26.3.1 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

27 Using the Query Analyzer	263
27.1 Providing Query Analyzer Data	264
27.1.1 Using the MySQL Performance Schema	264
27.2 Query Response Time index (QRTi)	267
27.3 Query Analyzer User Interface	268
27.3.1 Getting Detailed Query Information	270
27.3.2 Using Graphs to Identify Queries	271
27.3.3 Filtering Query Analyzer Data	272
27.3.4 Query Analyzer Settings	273
27.3.5 Exporting Query Information	274

Chapter 27 Using the Query Analyzer

Table of Contents

27.1 Providing Query Analyzer Data	264
27.1.1 Using the MySQL Performance Schema	264
27.2 Query Response Time index (QRTi)	267
27.3 Query Analyzer User Interface	268
27.3.1 Getting Detailed Query Information	270
27.3.2 Using Graphs to Identify Queries	271
27.3.3 Filtering Query Analyzer Data	272
27.3.4 Query Analyzer Settings	273
27.3.5 Exporting Query Information	274

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 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 27.1, “Providing Query Analyzer Data”](#). For the user interface of the Query Analyzer, see [Chapter 27, Using the Query Analyzer](#) and [Section 27.3, “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 27.3, “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.

27.1 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`.

27.1.1 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 digested Performance Schema query limit is 1,024 characters in length, in that queries exceeding this length are truncated.

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.

27.2 Query Response Time index (QRTi)

QRTi stands for "Query Response Time index". It is a "quality of service" measurement for each query, and 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 27.1 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.

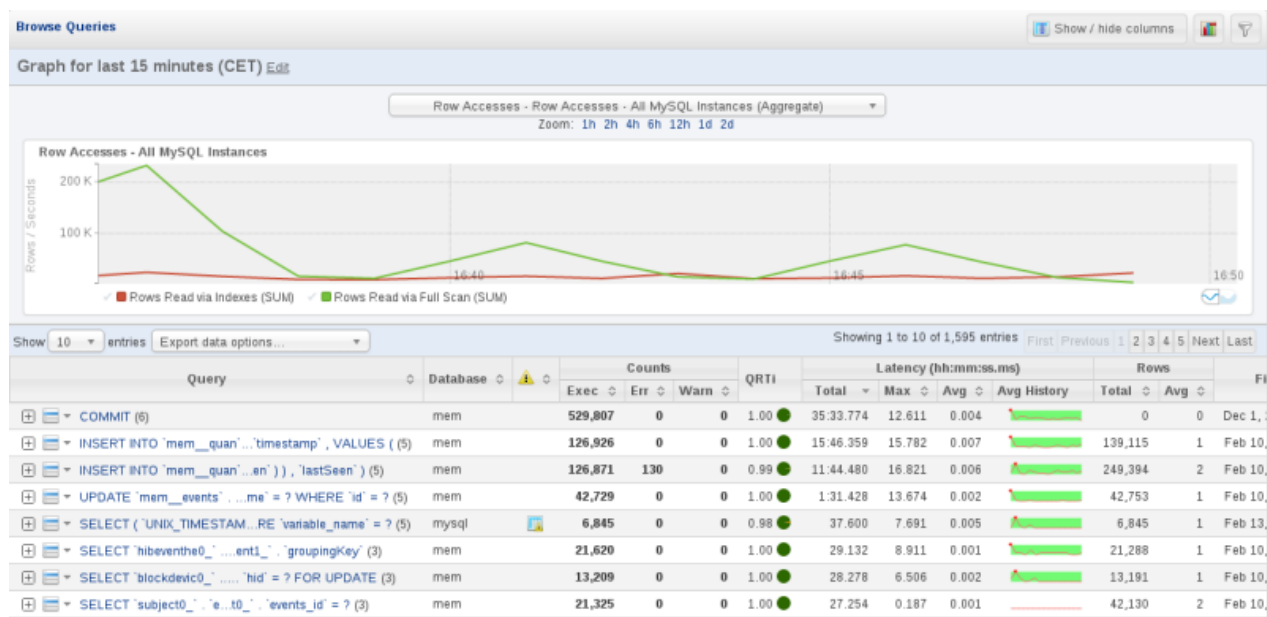
So when doing query optimization, you want to start with the ones that have a QRTi visual pie chart that is 100% red, which means that they also have an actual QRTi value of 0. This means that **all** executions of that query took longer than the acceptable time frame (400ms by default). You can then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

27.3 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 27.1 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 27.3.3, "Filtering Query Analyzer Data"](#). To export the query information as a **.csv** file, see [Section 27.3.5, "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 27.2, "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 27.3.3, “Filtering Query Analyzer Data”](#).

27.3.1 Getting Detailed Query Information

Click on an individual query to see more detailed information about the individual query in a pop-up window. 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:

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:

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

In addition to the full query, with data, that was executed, the tab shows the execution time, data, user, thread ID, client host and execution host for the given query.

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

- The **Explain Query** tab:

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



Important

Explain 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:

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.

27.3.2 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.

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 [Section 18.1, “All Timeseries Graphs”](#).

27.3.3 Filtering Query Analyzer Data

You can filter the queries shown within the Query Analyzer table by using the form at the top of the table. The different fields of the form are used to specify the parameters for the filter process. Once you have specified a filter, all the queries and related statistics shown within the Query Analyzer table are displayed in relation to the filter settings. For example, by default, the filter settings show the queries for the last 30 minutes. All the statistics shown are relative to the last 30 minutes, including average, maximum and execution counts.

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

The basic filter options are:

- **Statement Text** and **Value** support text searching of the normalized query. For the search type you can specify either a basic text match (**Contains**), or a regular expression match (**Regex**). In addition to the basic text match, you can also search for a query that does not contain a particular string. For regular expression searches, you can specify whether the regular expression should match, or not match (negative regexp) the queries. Regular expressions are parsed using the standard MySQL [REGEXP \(\)](#) function. For more information, see [Regular Expressions](#).



Note

The search is performed against the canonical version of the query. You cannot search against specific text or values within the parameters of the query itself.

- **Statement Type**: Limits the search to statements of a particular type ([SELECT](#), [LITERAL](#), etc.).
- **DB Name**: Limits the queries to those executed within a specific database. The database match is performed using the [LIKE](#) match from the MySQL database, hence you can use the `%` and `_` characters to multiple and single character matches. For more information, see [Pattern Matching](#).
- The **Time Range** menu selects whether the time selection for filtering should be based on the time **interval** (only queries recorded within the displayed time period are shown, using the **Hours** and **Minutes** pop-up), or whether the selection should be based on a time period (**From/To**), where you can select the time range to be displayed.

Using the **Interval** mode shows queries within the given time period from the point the graph was updated. For example, if you select 30 minutes, then the queries shown were captured within the last 30 minutes. If you updated the display at 14:00, then the queries displayed would have been captured between 13:30 and 14:00. Using interval mode limits the timespan for the filter selection to a maximum of 23 hours and 59 minutes.

Using the **From/To** time range lets you show queries between specific dates and times. Using this mode you can show only the queries received during a specific time span, and you can display the query history for a much longer time period, for as long as you have been recording query analysis information.

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

To use the advanced filtering techniques, click **show advanced**. This provides additional filters:

- **Notices:** Filters on the notices column, allowing you to filter the list to show only the queries that did not raise a notice, indicated a full table scan, or indicated that a bad index was used.
- Two column filters are provided, which allow you to filter the queries based on specific values within any of the columns shown in the Query Analyzer report list.

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

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

- The **View** selection determines whether the information should be returned on a group basis, where an aggregate of the same query executed on all monitored servers is shown, or on a **Server** basis, where queries are summarized by individual server. If the latter option has been selected, the table includes an additional column showing the server.

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

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

If you want to make the current filter options the default when viewing the **Query Analyzer** page, click the **Save As Default** button. The settings are saved for the current user only.

27.3.4 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.

Like with any Advisor, this may be set globally, or for a group or particular MySQL server.

Three 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 27.3.1, “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.

- **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 27.2, “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.

27.3.5 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** pop-up, 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 Component Reference	279
A.1 MySQL Enterprise Service Manager Reference	279
A.1.1 Log Files for the MySQL Enterprise Service Manager	279
A.1.2 The Management Information Base (MIB) File	279
A.1.3 The <code>config.properties</code> file	279
A.2 MySQL Enterprise Monitor Agent Reference	284
A.2.1 Agent Log Files	284
B Managing the Inventory	287
B.1 The Inventory Page	287
B.2 Using the Inventory Page	288
C MySQL Enterprise Monitor Frequently Asked Questions	289
D MySQL Enterprise Monitor Support	295
D.1 Diagnostics Report	295

Appendix A MySQL Enterprise Monitor Component Reference

Table of Contents

A.1 MySQL Enterprise Service Manager Reference	279
A.1.1 Log Files for the MySQL Enterprise Service Manager	279
A.1.2 The Management Information Base (MIB) File	279
A.1.3 The <code>config.properties</code> file	279
A.2 MySQL Enterprise Monitor Agent Reference	284
A.2.1 Agent Log Files	284

A.1 MySQL Enterprise Service Manager Reference

A.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 A.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

On all operating systems, the Apache/Tomcat, and Repository directories contain both access and error files.

A.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 A.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

A.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 A.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-dbcp/configuration.html>.

Table A.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
<code>dbpool.default.maxWaitMillis</code>	string	30 seconds
<code>dbpool.default.minEvictableIdleTimeMillis</code>	string	15 seconds

Property Name	Property Type	Default
<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	00:01:00

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	false

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	6 hours

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	5

- `dbpool.ui.maxActive(15)`

Permitted Values	Type	integer
	Default	15

- `dbpool.ui.minIdle(0)`

Permitted Values	Type	integer
	Default	0

- `dbpool.ui.maxIdle(5)`

Permitted Values	Type	integer
------------------	------	---------

	Default	5
--	----------------	---

- `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

A.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.

A.2.1 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/>.

Appendix B Managing the Inventory

Table of Contents

B.1 The Inventory Page	287
B.2 Using the Inventory Page	288

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 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

- [Security](#)
- [General Usage](#)
- [MySQL Monitor](#)
- [MySQL Query Analyzer](#)

Security

Questions

- [C.1](#): If I upgrade to 3.1, what happens to the users defined in earlier versions?

Questions and Answers

C.1: If I upgrade to 3.1, what happens to the users defined in earlier versions?

All users defined in earlier versions are mapped to the default roles introduced in Access Control Lists in MySQL Enterprise Monitor 3.1. The user names are retained but their permissions are defined separately in default roles. All pre-existing users are automatically mapped to the default roles.

For example, if User1 is defined as a dba in MySQL Enterprise Monitor 3.0.x, User1 is created in MySQL Enterprise Monitor 3.1, but assigned to the [dba](#) Role. If User1 is defined as a dba, and granted both Query Analyzer permissions in 3.0.x, it is assigned to the default dba Role, and both Query Analyzer roles in 3.1.

General Usage

Questions

- [C.1](#): How do I find [Ignored](#) MySQL Instances? And how to I show them again?
- [C.2](#): In 2.3, the [agent-mgmt-hostname](#) contained the string "heartbeat" as the URLs path. Did this change?
- [C.3](#): How do I change the name of a server?
- [C.4](#): Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?
- [C.5](#): 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?
- [C.6](#): Why does monitoring a MySQL instance with FEDERATED tables cause extra connections, and decreased performance?

Questions and Answers

C.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.

C.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.

C.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.

C.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.

C.5: 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.

C.6: 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

- [C.1](#): What are the features and related benefits of the MySQL Enterprise Monitor?
- [C.2](#): What are the immediate benefits of implementing the MySQL Enterprise Monitor?
- [C.3](#): What are the long-term benefits of the MySQL Enterprise Monitor?
- [C.4](#): How is the MySQL Enterprise Monitor installed and deployed?
- [C.5](#): How is the Enterprise Monitor web application architected?
- [C.6](#): What makes MySQL Enterprise unique?

-
- **C.7:** What versions of MySQL are supported by the MySQL Enterprise Monitor?
 - **C.8:** What operating system platforms are supported by the MySQL Enterprise Monitor?
 - **C.9:** How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?

Questions and Answers

C.1: 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>.

C.2: 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

C.3: 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.

C.4: 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.

C.5: 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.

C.6: 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.

C.7: What versions of MySQL are supported by the MySQL Enterprise Monitor?

The MySQL Enterprise Monitor supports MySQL versions 5.1 and above.

C.8: 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](#).

C.9: 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

- [C.1](#): What is the MySQL Query Analyzer?
- [C.2](#): How is the MySQL Query Analyzer installed and enabled?
- [C.3](#): What overhead can I expect when the MySQL Query Analyzer is installed and enabled?
- [C.4](#): What are the main features and benefits of the MySQL Query Analyzer?
- [C.5](#): What are the typical use cases of the MySQL Query Analyzer?
- [C.6](#): What makes the MySQL Query Analyzer unique?
- [C.7](#): How can I get the MySQL Query Analyzer?
- [C.8](#): Does Query Analyzer work with MySQL Cluster?
- [C.9](#): Does Query Analyzer enable me to monitor the disk reads and writes during a query?
- [C.10](#): Does Query Analyzer handle prepared statements?
- [C.11](#): How much degradation in performance does mysql-proxy introduce?
- [C.12](#): Will the Query Analyzer work without any special setup?

Questions and Answers

C.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>

C.2: How is the MySQL Query Analyzer installed and enabled?

See [Section 27.1, “Providing Query Analyzer Data”](#).

C.3: What overhead can I expect when the MySQL Query Analyzer is installed and enabled?

The average overhead when in active collection mode is in the 15-20% range. In pass-thru mode the overhead is minimal, weighing in at 1-5% on most MySQL systems of average load.

C.4: What are the main features and benefits of the MySQL Query Analyzer?

For the complete features and benefits, visit the <http://www.mysql.com/products/enterprise/monitor-features.html>

C.5: 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.

C.6: 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.

C.7: 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/>

C.8: 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.

C.9: 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.

C.10: 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.

C.11: 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.

C.12: 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 27.1, "Providing Query Analyzer Data"](#).

Appendix D MySQL Enterprise Monitor Support

Table of Contents

D.1 Diagnostics Report	295
------------------------------	-----

This appendix describes the Diagnostics Report.

D.1 Diagnostics Report

This chapter describes the Diagnostics Report. If you intend to communicate with MySQL Enterprise Monitor support, it is strongly recommended you provide this report with your support request.

Introduction

To generate a diagnostic report file, select **Diagnostics Report** from the **Settings** menu. The information is provided as a time stamped Zip file (such as [support-20160115T2238.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 very useful for debugging the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent. When filling out a My Oracle Support (MOS) ticket include this report.

Diagnostics Report File Contents

- [audit.log](#): The Audit log file.
- [catalina.out](#): A Tomcat log file.
- [com.mysql.etools.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.

Index

Symbols

.NET connector plugin, 110

, 81

A

Administration Advisors, 171

32-Bit Binary Running on 64-Bit AMD Or Intel System, 172

Binary Log Debug Information Disabled, 172

Binary Logging Is Limited, 172

Binary Logging Not Enabled, 172

Binary Logging Not Synchronized To Disk At Each Write, 173

Binary Logs Automatically Removed Too Quickly, 173

Database May Not Be Portable Due To Identifier Case Sensitivity, 173

Event Scheduler Disabled, 173

General Query Log Enabled, 174

Host Cache Size Not Sufficient, 174

In-Memory Temporary Table Size Limited By

Maximum Heap Table Size, 174

InnoDB Status Truncation Detected, 175

InnoDB Strict Mode Is Off, 175

InnoDB Tablespace Cannot Automatically Expand, 175

InnoDB Transaction Logs Not Sized Correctly, 175

Multiple Threads Used When Repairing MyISAM Tables, 176

MySQL Server No Longer Eligible For Oracle Premier Support, 176

Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 176

No Value Set For MyISAM Recover Options, 176

Table Cache Set Too Low For Startup, 176

Time Zone Data Not Loaded, 177

Warnings Not Being Logged, 177

--adminpassword option, 65

--adminuser option, 65

Advisors, 207

32-Bit Binary Running on 64-Bit AMD Or Intel System, 172

Account Has An Overly Broad Host Specifier, 200

Account Has Global Privileges, 200

Account Has Old Insecure Password Hash, 201

Account Has Strong MySQL Privileges, 201

Account Requires Unavailable Authentication Plug-ins, 201

Agent Health Advisor, 207

Attempted Connections To The Server Have Failed, 178

AUTO_INCREMENT Field Limit Nearly Reached, 195

Average Statement Execution Time Advisor, 216

Binary Log Checksums Disabled, 190

Binary Log Debug Information Disabled, 172

Binary Log File Count Exceeds Specified Limit, 191

Binary Log Row Based Images Excessive, 191

Binary Log Space Exceeds Specified Limit, 191

Binary Log Usage Exceeding Disk Cache Memory Limits, 185

Binary Logging Is Limited, 172

Binary Logging Not Enabled, 172

Binary Logging Not Synchronized To Disk At Each Write, 173

Binary Logs Automatically Removed Too Quickly, 173

Cluster Data Node Data Memory Getting Low, 180

Cluster Data Node Has Been Restarted, 180

Cluster Data Node Index Memory Getting Low, 180

Cluster Data Node Redo Buffer Space Getting Low, 180

Cluster Data Node Redo Log Space Getting Low, 180

Cluster Data Node Undo Buffer Space Getting Low, 180

Cluster Data Node Undo Log Space Getting Low, 180

Cluster Data Nodes Not Running, 181

Cluster DiskPageBuffer Hit Ratio Is Low, 181

Cluster Has Stopped, 181

CPU Utilization Advisor, 213

Database May Not Be Portable Due To Identifier Case Sensitivity, 173

Duplicate MySQL Server UUID, 212

Event Scheduler Disabled, 173

Excessive Disk Temporary Table Usage Detected, 185

Excessive Number of Locked Processes, 185

Excessive Number of Long Running Processes, 185

Excessive Number of Long Running Processes Locked, 186

Excessive Percentage Of Attempted Connections To The Server Have Failed, 178

Filesystem Free Space Advisor, 214

Flush Time Set To Non-Zero Value, 186

General Query Log Enabled, 174

Host Cache Size Not Sufficient, 174

HTTP Server KeyStore's Certificate About to Expire Advisor, 212

HTTP Server Performance, 183

In-Memory Temporary Table Size Limited By

Maximum Heap Table Size, 174

Indexes Not Being Used Efficiently, 186

InnoDB Buffer Cache Has Sub-Optimal Hit Rate, 181

InnoDB Buffer Pool Writes May Be Performance Bottleneck, 186

InnoDB Flush Method May Not Be Optimal, 186

InnoDB Log Buffer Flushed To Disk After Each Transaction, 187

InnoDB Log Waits May Be Performance Bottleneck, 187

InnoDB Not Using Newest File Format, 187

InnoDB Status Truncation Detected, 175

InnoDB Strict Mode Is Off, 175

InnoDB Tablespace Cannot Automatically Expand, 175

InnoDB Transaction Logs Not Sized Correctly, 175

Insecure Password Authentication Option Is Enabled, 201

Insecure Password Generation Option Is Enabled, 201

Key Buffer Size May Not Be Optimal For Key Cache, 182

LOCAL Option Of LOAD DATA Statement Is Enabled, 202

Master Not Verifying Checksums When Reading From Binary Log, 192

Maximum Connection Limit Nearing Or Reached, 179

Multiple Threads Used When Repairing MyISAM Tables, 176

MyISAM Concurrent Insert Setting May Not Be Optimal, 187

MyISAM Indexes Found with No Statistics, 197

MySQL Agent Memory Usage Excessive, 177

MySQL Agent Not Reachable, 177

MySQL Availability, 179

MySQL Enterprise Backup Health Advisor, 210

MySQL Process Discovery Advisor, 211

MySQL Server Has Been Restarted, 179

MySQL Server No Longer Eligible For Oracle Premier Support, 176

Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 176

No Value Set For MyISAM Recover Options, 176

Non-root User Has DB, Table, Or Index Privileges On All Databases, 202

Non-root User Has GRANT Privileges On All Databases, 202

Non-root User Has Server Admin Privileges, 202

Object Changed: Database Has Been Altered, 196

Object Changed: Database Has Been Created, 196

Object Changed: Database Has Been Dropped, 196

Object Changed: Function Has Been Created, 196

Object Changed: Function Has Been Dropped, 196

Object Changed: Index Has Been Created, 197

Object Changed: Index Has Been Dropped, 197

Object Changed: Table Has Been Altered, 198

Object Changed: Table Has Been Created, 198

Object Changed: Table Has Been Dropped, 198

Object Changed: User Has Been Dropped, 199

Object Changes Detected, 197

Policy-Based Password Validation Does Not Perform Dictionary Checks, 203

Policy-Based Password Validation Is Weak, 203

Policy-Based Password Validation Not Enabled, 203

Prepared Statements Not Being Closed, 188

Prepared Statements Not Being Used Effectively, 188

Privilege Alterations Detected: Privileges Granted, 203

Privilege Alterations Detected: Privileges Revoked, 203

Privilege Alterations Have Been Detected, 204

Query Analysis Reporting, 217

Query Cache Has Sub-Optimal Hit Rate, 182

Query Cache Is Excessively Fragmented, 188

Query Cache Potentially Undersized, 182

Query Pileup Advisor, 216

Replication Configuration Advisor, 191

Replication Status Advisor, 191

Root Account Can Login Remotely, 204

Root Account Without Password, 204

Server Contains Default "test" Database, 204

Server Has Accounts Without A Password, 204

Server Has Anonymous Accounts, 205

Server Has No Locally Authenticated Root User, 205

Server Includes A Root User Account, 205

Server-Enforced Data Integrity Checking Disabled, 197

Server-Enforced Data Integrity Checking Not Strict, 198

SHA-256 Password Authentication Not Enabled, 204

Slave Detection Of Network Outages Too High, 192

Slave Execution Position Too Far Behind Read Position, 192

Slave Has Login Accounts With Inappropriate Privileges, 192

Slave Master Info/Relay Log Info Not Crash Safe, 193

Slave Not Configured As Read Only, 193

Slave Not Verifying Checksums When Reading From Relay Log, 193

Slave Relay Log Space Is Very Large, 193, 193

Slave SQL Processing Not Multi-Threaded, 194

Slave SQL Thread Reading From Older Relay Log Than I/O Thread, 194

Slave Too Far Behind Master, 194

Slave Without REPLICATION SLAVE Accounts, 194

SQL Statement Generates Warnings or Errors, 217

Support Diagnostics, 183

Symlinks Are Enabled, 205

sys Schema Install Advisor, 212

Table Cache Not Optimal, 182

Table Cache Set Too Low For Startup, 176

Table Lock Contention Excessive, 188

Tables Found with No Primary or Unique Keys, 198

Thread Cache Not Enabled, 189

Thread Cache Size May Not Be Optimal, 182

Thread Pool Stall Limit Too Low, 189

Thread Pooling Not Enabled, 189

Time Zone Data Not Loaded, 177

- Too Many Concurrent Queries Running, 189
- User Has Rights To Database That Does Not Exist, 206
- User Has Rights To Table That Does Not Exist, 206
- Users Can View All Databases On MySQL Server, 206
- Warnings Not Being Logged, 177
- Wrong Version Agent Tracker, 184
- advisors
 - creating, 245, 250
- agent
 - installation, 33
 - troubleshooting, 46
- Agent Advisors, 177
 - Agent Health Advisor, 207
 - MySQL Agent Memory Usage Excessive, 177
 - MySQL Agent Not Reachable, 177
- agent_autocreate option, 70
- agent_installtype option, 72
- agentpassword option, 67
- agentservicename option, 72
- agentuser option, 67
- Aggregator
 - PHP connector, 101
- Apple OS X, 42, 279
- architecture, 5
- Availability Advisors, 178
 - Attempted Connections To The Server Have Failed, 178
 - Excessive Percentage Of Attempted Connections To The Server Have Failed, 178
 - Maximum Connection Limit Nearing Or Reached, 178
 - MySQL Availability, 179
 - MySQL Server Has Been Restarted, 179

B

- backup
 - restore, 51
- Backup Advisors
 - MySQL Enterprise Backup Health Advisor, 210
- backupdir option, 61
- blackout period, 256
- browsers, 23

C

- checkmysqlhost option, 67
- Cluster Advisors, 179
 - Cluster Data Node Data Memory Getting Low, 179
 - Cluster Data Node Has Been Restarted, 180
 - Cluster Data Node Index Memory Getting Low, 180
 - Cluster Data Node Redo Buffer Space Getting Low, 180
 - Cluster Data Node Redo Log Space Getting Low, 180

- Cluster Data Node Undo Buffer Space Getting Low, 180
- Cluster Data Node Undo Log Space Getting Low, 180
- Cluster Data Nodes Not Running, 181
- Cluster DiskPageBuffer Hit Ratio Is Low, 181
- Cluster Has Stopped, 181
- config.properties file, 279
- configuration files, 279
- configuration utilities
 - agent, 83
 - agent.bat, 83
 - agent.sh, 83
 - config.bat, 81
 - config.sh, 81
 - service manager, 81
- Connector/J, 105
- Connector/Net, 110
- createBackup option, 67
- createDataBackup option, 61
- custom data collection
 - customizing advisors, 252

D

- data_collection_interval option, 281
- dbhost option, 66
- dbname option, 66
- dbpool.default.initialSize option, 283
- dbpool.default.maxActive option, 283
- dbpool.default.maxIdle option, 283
- dbpool.default.maxWaitMillis option, 283
- dbpool.default.minEvictableIdleTimeMillis option, 284
- dbpool.default.minIdle option, 283
- dbpool.default.timeBetweenEvictionRunsMillis option, 283
- dbpool.ui.initialSize option, 282
- dbpool.ui.maxActive option, 282
- dbpool.ui.maxIdle option, 282
- dbpool.ui.maxWaitMillis option, 283
- dbpool.ui.minEvictableIdleTimeMillis option, 283
- dbpool.ui.minIdle option, 282
- dbpool.ui.timeBetweenEvictionRunsMillis option, 283
- dbport option, 66
- debuglevel option, 63, 68
- debugtrace option, 62, 68
- diagnostic report, 295
- Diagnostics Report, 295

E

- expressions, 249

F

- FAQs, 289
- firewall issues, 45

--forceRestart option, 66

G

--generalpassword option, 71

--generaluser option, 71

graphs

creating, 245, 251

H

--help option, 61, 73

I

--ignore-old-proxy-aggr option, 72

InnoDB Buffer Pool Usage Report, 148

installation

agent, 33

backup, 51

post-install tasks, 53

service manager, 25

unattended, 59

--installdir option, 63, 68

--installer option, 63

--installer-language option, 68

internal_perf_enable option, 281

internal_perf_server_id option, 282

J

Java connector plugin, 105

Java VM

installation, 25

L

--limitedpassword option, 71

--limiteduser option, 71

Linux, 43, 279

locale, 118

log file

MySQL Enterprise Service Manager, 279

M

Mac OS X, 42, 279

--managerhost option, 68

--managerport option, 69

Memory Usage Advisors, 181

InnoDB Buffer Cache Has Sub-Optimal Hit Rate, 181

Key Buffer Size May Not Be Optimal For Key Cache, 182

Query Cache Has Sub-Optimal Hit Rate, 182

Query Cache Potentially Undersized, 182

Table Cache Not Optimal, 182

Thread Cache Size May Not Be Optimal, 182

MIB file, 279

--mode option, 62, 69

monitor

installation, 25

Monitoring and Support Advisors

HTTP Server KeyStore's Certificate About to Expire Advisor, 212

sys Schema Install Advisor, 212

Monitoring and Support Services Advisors, 183

Duplicate MySQL Server UUID, 212

HTTP Server Performance, 183

MySQL Process Discovery Advisor, 211

Support Diagnostics, 183

Wrong Version Agent Tracker, 184

MySQL Server

installation, 25

mysql-monitor-agent.log file, 284

--mysqlconnectiongroup option, 72

--mysqlconnmethod option, 69

mysqlenterprise.* options, 101, 105

--mysqlhost option, 70

--mysql-identity-source option, 64

--mysqlpassword option, 70

--mysqlport option, 70

--mysqlsocket option, 70

--mysqluser option, 70

--mysql_installation_type option, 65

--mysql_ssl option, 64

N

notify_thread_pool_size option, 281

O

Operating System Advisors, 184

CPU Utilization Advisor, 213

Filesystem Free Space Advisor, 214

--optionfile option, 62, 71

OS X, 42, 279

overview, 5

Overview Dashboard, 123

P

Performance Advisors, 184

Binary Log Usage Exceeding Disk Cache Memory Limits, 185

Excessive Disk Temporary Table Usage Detected, 185

Excessive Number of Locked Processes, 185

Excessive Number of Long Running Processes, 185

Excessive Number of Long Running Processes

Locked, 186

Flush Time Set To Non-Zero Value, 186

Indexes Not Being Used Efficiently, 186

InnoDB Buffer Pool Writes May Be Performance

Bottleneck, 186

InnoDB Flush Method May Not Be Optimal, 186

- InnoDB Log Buffer Flushed To Disk After Each Transaction, 187
- InnoDB Log Waits May Be Performance Bottleneck, 187
- InnoDB Not Using Newest File Format, 187
- MyISAM Concurrent Insert Setting May Not Be Optimal, 187
- Prepared Statements Not Being Closed, 188
- Prepared Statements Not Being Used Effectively, 188
- Query Cache Is Excessively Fragmented, 188
- Table Lock Contention Excessive, 188
- Thread Cache Not Enabled, 189
- Thread Pool Stall Limit Too Low, 189
- Thread Pooling Not Enabled, 189
- Too Many Concurrent Queries Running, 189
- performance schema, 264
- performance tuning, 75
- PHP connector plugin
 - Aggregator, 101
- ports
 - MySQL server for monitoring, 70
 - MySQL server for repository, 66
 - service manager, 64, 69
 - Tomcat, 64
 - Tomcat SSL, 64
- post-install tasks, 53

Q

- quanal.collect option, 282
- Query Analysis Advisors, 216
 - Average Statement Execution Time Advisor, 216
 - Query Analysis Reporting, 217
 - Query Pileup Advisor, 216
 - SQL Statement Generates Warnings or Errors, 217
- Query Analyzer, 263
 - .NET connector, 110
 - Java connector, 105
 - PHP connector, 101
 - supplying query data, 264
- Query Analyzer tab, 268

R

- Replication, 139
- Replication Advisors, 190
 - Binary Log Checksums Disabled, 190
 - Binary Log File Count Exceeds Specified Limit, 191
 - Binary Log Row Based Images Excessive, 191
 - Binary Log Space Exceeds Specified Limit, 191
 - Master Not Verifying Checksums When Reading From Binary Log, 192
 - Replication Configuration Advisor, 191
 - Replication Status Advisor, 191
 - Slave Detection Of Network Outages Too High, 192

- Slave Execution Position Too Far Behind Read Position, 192
- Slave Has Login Accounts With Inappropriate Privileges, 192
- Slave Master Info/Relay Log Info Not Crash Safe, 193
- Slave Not Configured As Read Only, 193
- Slave Not Verifying Checksums When Reading From Relay Log, 193
- Slave Relay Log Space Is Very Large, 193, 193
- Slave SQL Processing Not Multi-Threaded, 194
- Slave SQL Thread Reading From Older Relay Log Than I/O Thread, 194
- Slave Too Far Behind Master, 194
- Slave Without REPLICATION SLAVE Accounts, 194
- repository
 - database name, 66
- restartImmediately option, 67
- restore
 - backup, 51
- restoring
 - backup, 51
- rules
 - blackout periods, 256
 - creating, 245
 - variable substitution, 248

S

- Schema Advisors, 195
 - AUTO_INCREMENT Field Limit Nearly Reached, 195
 - MyISAM Indexes Found with No Statistics, 197
 - Object Changed: Database Has Been Altered, 196
 - Object Changed: Database Has Been Created, 196
 - Object Changed: Database Has Been Dropped, 196
 - Object Changed: Function Has Been Created, 196
 - Object Changed: Function Has Been Dropped, 196
 - Object Changed: Index Has Been Created, 196
 - Object Changed: Index Has Been Dropped, 197
 - Object Changed: Table Has Been Altered, 198
 - Object Changed: Table Has Been Created, 198
 - Object Changed: Table Has Been Dropped, 198
 - Object Changed: User Has Been Dropped, 199
 - Object Changes Detected, 197
 - Server-Enforced Data Integrity Checking Disabled, 197
 - Server-Enforced Data Integrity Checking Not Strict, 197
 - Tables Found with No Primary or Unique Keys, 198
- Security Advisors, 199, 217
 - Account Has An Overly Broad Host Specifier, 200
 - Account Has Global Privileges, 200
 - Account Has Old Insecure Password Hash, 200
 - Account Has Strong MySQL Privileges, 201

- Account Requires Unavailable Authentication Plug-ins, 201
- Insecure Password Authentication Option Is Enabled, 201
- Insecure Password Generation Option Is Enabled, 201
- LOCAL Option Of LOAD DATA Statement Is Enabled, 202
- MySQL Enterprise Audit Plugin, 217
- Non-root User Has DB, Table, Or Index Privileges On All Databases, 202
- Non-root User Has GRANT Privileges On All Databases, 202
- Non-root User Has Server Admin Privileges, 202
- Policy-Based Password Validation Does Not Perform Dictionary Checks, 202
- Policy-Based Password Validation Is Weak, 203
- Policy-Based Password Validation Not Enabled, 203
- Privilege Alterations Detected: Privileges Granted, 203
- Privilege Alterations Detected: Privileges Revoked, 203
- Privilege Alterations Have Been Detected, 204
- Root Account Can Login Remotely, 204
- Root Account Without Password, 204
- Server Contains Default "test" Database, 204
- Server Has Accounts Without A Password, 204
- Server Has Anonymous Accounts, 205
- Server Has No Locally Authenticated Root User, 205
- Server Includes A Root User Account, 205
- SHA-256 Password Authentication Not Enabled, 204
- Symlinks Are Enabled, 205
- User Has Rights To Database That Does Not Exist, 205
- User Has Rights To Table That Does Not Exist, 206
- Users Can View All Databases On MySQL Server, 206
- service manager
 - database name, 66
 - installation, 25
- services
 - starting and stopping, 29
- SNMP traps, 279
- sql_mode, 44
- SSH tunnelling, 45
- SSL, 64
- starting
 - MySQL Enterprise Monitor service, 30
 - MySQL Enterprise Monitor services, 29
- stopping
 - MySQL Enterprise Monitor service, 30
 - MySQL Enterprise Monitor services, 29
- support files
 - diagnostic report, 295
- supportReport.retention.minutes option, 282
- system-size option, 63

T

- thread_pool_size option, 281
- thresholds, 248
- timezone, 118
- Tomcat
 - installation, 25
 - starting and stopping, 30
- tomcatport option, 64
- tomcatsslport option, 64
- troubleshooting, 46
- tuning, 75

U

- ui.javascript.useClientSideStorage option, 282
- unattended installation, 59
- unattendedmodeui option, 65, 71
- Unix, 43, 279

V

- variable substitution, 248
- version option, 61, 72

W

- Web browsers, 23
- Wiki markup, 250
- Windows, 30, 40, 279