

MySQL Enterprise Monitor 8.0.39 Manual

Abstract

This manual documents MySQL Enterprise Monitor version 8.0.39.

End of Life Announcement. MySQL Enterprise Monitor will be end of life and deprecated with obsolescence as of January 1, 2025. For additional information, see [Chapter 1, *MySQL Enterprise Monitor: End of Life Notice*](#)

For notes detailing the changes in each release, see the [MySQL Enterprise Monitor 8.0 Release Notes](#).

For legal information, including licensing information, see the [Preface and Legal Notices](#).

For help with using MySQL, please visit the [MySQL Forums](#), where you can discuss your issues with other MySQL users.

Document generated on: 2024-07-27 (revision: 8639)

Table of Contents

Preface and Legal Notices	ix
I Getting Started with MySQL Enterprise Monitor	1
1 MySQL Enterprise Monitor: End of Life Notice	5
2 MySQL Enterprise Monitor Introduction and Architecture	7
2.1 MySQL Enterprise Monitor Component Overview	7
2.2 MySQL Enterprise Monitor Agent	8
2.3 MySQL Enterprise Service Manager	9
3 What's New	13
II Installing MySQL Enterprise Monitor	21
4 Prerequisites	25
4.1 Installer Files	25
4.2 Prerequisites	25
4.2.1 System Requirements	25
4.2.2 Supported Platforms	26
4.2.3 Requirements for Monitoring	27
4.2.4 MySQL Enterprise Monitor Repository	28
4.3 Credentials Required for Installation	30
4.3.1 Existing Users	30
4.3.2 Users Created During Installation	31
4.3.3 Users Created on First Log-in	31
4.4 Supported Browsers	31
5 Service Manager Installation	33
5.1 MySQL Enterprise Monitor Installation Types	33
5.2 MySQL Enterprise Service Manager Graphical Installation Wizard	34
5.3 Text-Based Installation	36
5.4 Starting/Stopping MySQL Enterprise Monitor Services	37
5.5 MySQL Enterprise Service Manager Configuration Settings	38
6 Monitor Agent Installation	41
6.1 General Agent Related Notes	41
6.2 Creating MySQL User Accounts for the Monitor Agent	42
6.3 Java Considerations on UNIX-like Platforms	45
6.4 Monitoring Agent Graphical Installation Wizard	45
6.5 Starting/Stopping MySQL Enterprise Monitor Agent	48
6.5.1 Starting/Stopping the Agent on Windows	48
6.5.2 Starting/Stopping the Agent on macOS	49
6.5.3 Starting/Stopping the Agent on Unix	50
6.5.4 sql_mode	51
6.6 Monitoring Multiple MySQL Servers	51
6.7 Configuring an Agent to Monitor a Remote MySQL Server	51
6.8 Monitoring Outside the Firewall with an SSH Tunnel	52
6.9 HTTP Connection Timeout	52
6.10 Troubleshooting the Agent	52
6.11 Agent Backlog	54
7 Upgrading MySQL Enterprise Monitor Installations	55
7.1 General considerations when upgrading MySQL Enterprise Monitor	55
7.2 Upgrading MySQL Enterprise Monitor	55
7.3 Restoring from Backup	57
8 Post-installation Considerations	59
8.1 General Considerations	59
8.2 Installing SSL Certificates	60
8.3 Backup the Repository	63

8.4 Changing an SSH Host Key	64
9 Unattended Installation Reference	67
9.1 Unattended Installation	67
9.1.1 Performing an Unattended Installation	67
9.1.2 MySQL Enterprise Service Manager Options	68
9.1.3 MySQL Enterprise Monitor Agent Options	74
10 Performance Tuning MySQL Enterprise Monitor	83
10.1 Tuning Memory	83
10.2 Tuning CPU	84
10.3 Tuning Apache Tomcat Threads	85
10.4 Tuning Agent Memory Requirements	85
11 Configuration Utilities	87
11.1 Service Manager Configuration Utilities	87
11.2 Agent Configuration Utility	89
12 Uninstalling MySQL Enterprise Monitor	93
12.1 Windows Platforms	93
12.2 macOS Platforms	94
12.3 Unattended Uninstallations	95
III Configuring MySQL Enterprise Service Manager	97
13 Configuring MySQL Enterprise Service Manager	101
14 User Interface	103
14.1 Initial Log-In	103
14.2 Setting the Timezone and Locale	104
14.3 Menus and Toolbars	104
14.3.1 Target Selection	104
14.3.2 Navigation Menus	105
14.3.3 User Menu	107
14.3.4 Status Summary	108
14.3.5 System and User-defined Filters	108
14.3.6 Configuration Views	109
15 Global Settings	111
15.1 Server Locale	111
15.2 Server Hostname	111
15.3 Customize MySQL Server Name	111
15.4 Data Purge Behavior	113
15.5 HTTP Proxy Settings	114
15.6 External Authentication	114
16 Manage Instances	119
16.1 Manage Instances Controls	119
16.2 Monitored Instance Details	121
16.3 Adding Instances	123
16.3.1 Adding a MySQL Instance	123
16.3.2 Adding Multiple MySQL Instances	127
16.4 Monitoring Amazon RDS	127
16.5 Filtering MySQL Instances	128
17 Asset Management	131
17.1 Agents View	131
17.2 Hosts View	132
17.3 NDB Clusters View	134
17.4 Deleting Assets	135
17.5 Deleting Assets from the Command Line	138
18 Managing Groups of Instances	145
19 Advisors	149
19.1 Manage Advisors Page	149

19.2	Advisor Types	153
19.3	Advisor Thresholds	154
19.4	Advisor Schedules	156
20	Expression-Based Advisor Reference	157
20.1	Administration Advisors	157
20.2	Agent Advisors	163
20.3	Availability Advisors	163
20.4	NDB Cluster Advisors	165
20.5	Memory Usage Advisors	166
20.6	Monitoring and Support Services Advisors	168
20.7	Operating System Advisors	169
20.8	Performance Advisors	169
20.9	Replication Advisors	175
20.10	Schema Advisors	180
20.11	Security Advisors	184
21	GUI-Based Advisor Reference	191
21.1	Agent Health Advisor	191
21.2	MySQL Enterprise Backup Health Advisor	194
21.3	MySQL Process Discovery Advisor	194
21.4	Duplicate Host Identity	196
21.5	Duplicate MySQL Server UUID	196
21.6	sys Schema Install Advisor	196
21.7	CPU Utilization Advisor	196
21.8	Filesystem Free Space Advisor	198
21.9	MySQL Process	200
21.10	Query Analysis Advisors	200
21.11	Group Replication Advisors	202
21.12	Security Advisors	204
21.13	NDB Cluster GUI Advisors	206
22	Event Handlers	211
22.1	Event Handlers	211
22.1.1	Event Handlers	211
22.1.2	Event Handlers Page	211
22.2	Creating Event Handlers	217
22.2.1	Event Action Log	220
22.2.2	Suspending an Event Handler	220
23	Access Control	221
23.1	Users and Roles	221
23.2	Permissions	221
23.3	Monitored Assets Permissions	222
23.3.1	Server Group & MySQL NDB Cluster	223
23.3.2	MySQL Instances	223
23.3.3	MySQL NDB Cluster Permissions	225
23.4	Monitoring Services	225
23.5	MySQL Enterprise Monitor	226
23.6	Default Users and Roles	228
23.7	Creating Users and Roles	229
24	Access Control - Best Practices	231
24.1	Open Permission Sets	232
24.2	Strict Permission Set	233
IV	Using MySQL Enterprise Monitor	239
25	Overview	243
25.1	Overview Sparkline Graphs	244
25.2	Database Availability	247

25.3 General Database Statistics	248
26 Events	249
26.1 Events	249
27 Enterprise Backup Dashboard	253
27.1 Backup Overview	253
27.2 Backup Details	255
28 Replication Dashboard	259
28.1 Replication Overview	259
28.2 Replication Dashboard Tabs	260
28.3 Replication Logging	265
29 Topology	267
30 Reports and Graphs	271
30.1 All Timeseries Graphs	271
30.1.1 Graph Controls	271
30.1.2 Graph Types	272
30.1.3 Timeseries Data Aggregation	273
30.2 Database File I/O and Lock Waits	273
30.2.1 sys Schema	273
30.2.2 Database File I/O Graphs and Reports	274
30.2.3 Lock Waits Report	277
30.3 Processes Report	278
30.4 Table Statistics	278
30.5 User Statistics	280
30.6 Memory Usage Report	281
30.7 InnoDB Buffer Pool Usage	282
30.8 NDB Cluster Memory Usage	283
31 Customizing MySQL Enterprise Monitor	285
31.1 Creating Advisors and Rules	285
31.1.1 Creating Advisors	285
31.1.2 Overview of Graph Creation	286
31.1.3 Overview of Advisor Creation	286
31.1.4 Variables	288
31.1.5 Thresholds	288
31.1.6 Using Strings	289
31.1.7 Wiki Format	289
31.1.8 Creating a New Advisor: An Example	290
31.1.9 Creating a New Graph: An Example	291
31.2 Custom Data Collection	292
31.2.1 Custom.xml	292
31.2.2 Queries	293
31.2.3 Data Collection Attributes	294
31.3 Event Notification Blackout Periods	296
31.3.1 Scripting Blackouts	298
V Using the Query Analyzer	301
32 Query Analyzer View	305
32.1 Providing Query Analyzer Data	305
32.1.1 Using the MySQL Performance Schema	305
32.2 Query Response Time index (QRTi)	307
32.3 Query Analyzer User Interface	308
32.4 Detailed Query Information	311
32.5 Query Analyzer Configuration View	314
VI Appendices	323
A MySQL Enterprise Monitor Component Reference	327
A.1 MySQL Enterprise Service Manager Reference	327

A.1.1 MySQL Enterprise Service Manager Log Files	327
A.1.2 The Management Information Base (MIB) File	327
A.1.3 The <code>config.properties</code> file	328
A.2 MySQL Enterprise Monitor Agent Reference	331
A.2.1 Agent Log Files	331
B Managing the Inventory	333
B.1 The Inventory Page	333
B.2 Using the Inventory Page	334
C MySQL Enterprise Monitor Frequently Asked Questions	335
D MySQL Enterprise Monitor Support	341
D.1 Diagnostics Report	341
MySQL Enterprise Monitor Glossary	343
Index	355

Preface and Legal Notices

This manual documents the MySQL Enterprise Monitor version 8.0.39.

Licensing information—MySQL Enterprise Monitor. This product may include third-party software, used under license. See [MySQL Enterprise Monitor 8.0 License Information User Manual](#) 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, 2024, Oracle and/or its affiliates.

License Restrictions

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.

Warranty Disclaimer

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.

Restricted Rights Notice

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), 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 embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

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.

Trademark Notice

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

Intel and Intel Inside 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, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

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.

Use of This Documentation

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.

Access to Oracle Support for Accessibility

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.

Part I Getting Started with MySQL Enterprise Monitor

Table of Contents

1 MySQL Enterprise Monitor: End of Life Notice	5
2 MySQL Enterprise Monitor Introduction and Architecture	7
2.1 MySQL Enterprise Monitor Component Overview	7
2.2 MySQL Enterprise Monitor Agent	8
2.3 MySQL Enterprise Service Manager	9
3 What's New	13

Chapter 1 MySQL Enterprise Monitor: End of Life Notice



Important

MySQL Enterprise Monitor will be end of life and deprecated with obsolescence as of January 1, 2025.

After this date, MySQL Enterprise Monitor will no longer receive security updates, non-security updates, bug fixes, or online technical content updates. It will transition to the *Sustaining Support model*.

What to expect when MySQL Enterprise Monitor reaches the end of life (EOL):

- MySQL will cease all bug fix activities for the product
- MySQL will cease all security fix activities for the product
- MySQL will cease all new feature work for the product

Sustaining Support does not include:

- New program updates, fixes, security alerts, and critical patch updates
- New tax, legal, or regulatory updates
- New upgrade scripts
- Certification with new third-party products/versions
- 24 hour commitment and response guidelines for Severity 1 service requests as defined in "Section 9 - Severity Definitions" in the document titled "Oracle Software Technical Support Policies"
- Previously released fixes or updates that Oracle has withdrawn from publication. Older or existing published software bundles will remain available as archived content.

For the set of Oracle Technical Support Policies, visit: <https://www.oracle.com/support/policies.html>

For an explanation of the different support models (like Sustaining Support), visit: <https://www.oracle.com/support/lifetime-support/>

For customers that are currently using earlier versions of MySQL Enterprise Monitor, your options include:

- Use Enterprise Manager for MySQL. This is a free product for customers with a valid Oracle Support Contract. For more information, visit [Comprehensive Monitoring and Compliance Management for MySQL Databases using Enterprise Manager](#).
- Use the database monitoring capabilities of the OCI Database Management service. For MySQL on-premises customers, this is a paid feature that will be released soon. For more information, visit [Database Management for MySQL HeatWave](#).

Affected deployments

- MySQL Enterprise Monitor Service Manager for Linux x86 (64-bit)
- MySQL Enterprise Monitor Agent for Linux x86 (64-bit)
- MySQL Enterprise Monitor Service Manager for Windows x86 (64-bit)
- MySQL Enterprise Monitor Agent for Microsoft Windows x86 (64-bit)

- MySQL Enterprise Monitor Service Manager for Mac OS X x86 (64-bit)
- MySQL Enterprise Monitor Agent for Mac OS X x86 (64-bit)

Chapter 2 MySQL Enterprise Monitor Introduction and Architecture



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 <https://www.mysql.com/products/>.

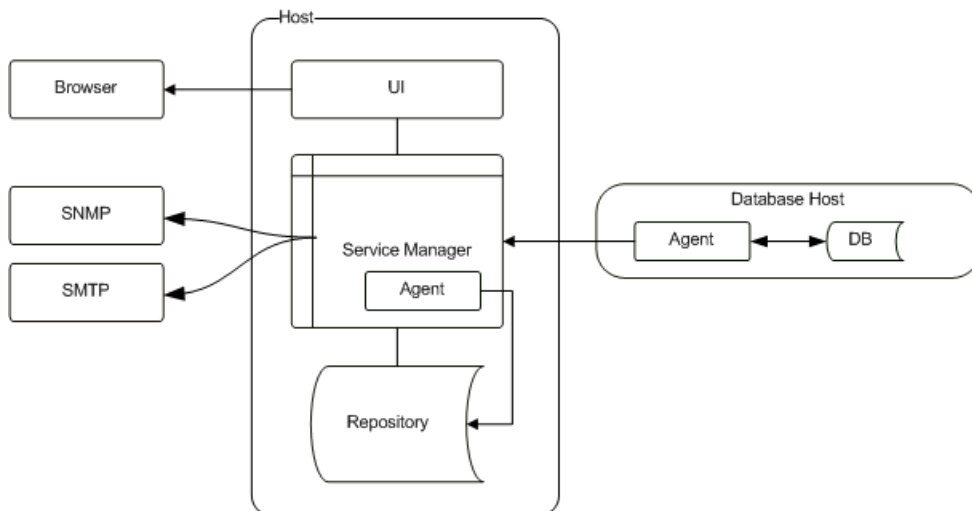
MySQL Enterprise Monitor is a companion product to MySQL Server that enables monitoring of MySQL instances and their hosts, notification of potential issues and problems, and advice on how to correct issues. MySQL Enterprise Monitor can monitor all types of installation, from a single MySQL instance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

2.1 MySQL Enterprise Monitor Component Overview

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

Figure 2.1 MySQL Enterprise Monitor Architecture

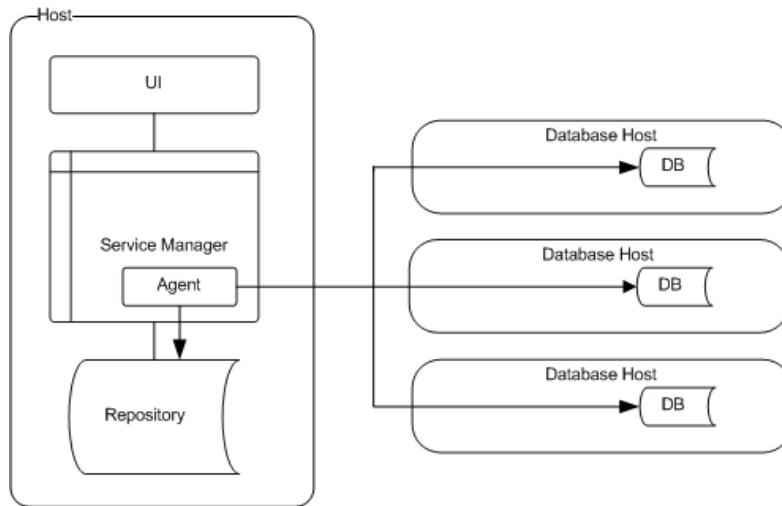


MySQL Enterprise Monitor has the following components:

- MySQL Enterprise Monitor Agent monitors the MySQL instances and hosts, and collects data according to a defined schedule. The collection data is sent to MySQL Enterprise Service Manager for analysis and presentation. MySQL Enterprise Monitor Agent is typically installed on the same host as the monitored server.
- MySQL Enterprise Service Manager analyzes, stores and presents the data collected by the agent. MySQL Enterprise Service Manager also contains a built-in agent which is used to monitor the repository.

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 MySQL Enterprise Service Manager's built-in Agent.

Figure 2.2 MySQL Enterprise Monitor Agentless Architecture



Important

For performance reasons, the agent-less 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.

2.2 MySQL Enterprise Monitor Agent

The Agent collects data from the monitored instance and host, and transmits that data to 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 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.

2.3 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 utilize 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:

- [Overview](#) provides an overview of the current health and status of all assets, or detailed information on the selection in the Target selection menus, a list of top critical and emergency events that should be handled, and graphs that relay database statistical information.
- The **Configuration** section enables you to customize the Advisors and Event Handling for your system. For example, this includes setting thresholds for Advisors, Users & Roles, Group configuration, and

email addresses to send alerts. This section also contains the [Instances](#) view, which lists all monitored MySQL instances, which can be analyzed, configured, and grouped and enables you to add and remove monitored instances.

- [Query Analyzer](#) enables analysis of problematic queries.
- [Replication](#) displays the details and health of your replication environment.
- The [Events](#) view lists the events generated by all monitored assets, which can be sorted and searched.
- **Metrics** view includes all graphs and reports with compiled data for your system, updated according to the assets selected in the target selector. The contents of the Metrics section are dependent on the selection made in the Target selectors. Some reports are only useful for NDB Cluster monitoring, or for Replication Groups, and are only displayed if one of those assets are selected.

MySQL Enterprise Advisors

Advisors filter and evaluate the information broadcast by the Monitoring Agents and present it to the Events view 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.
- **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.
- **NDB Cluster:** Checks the status of the monitored MySQL NDB Cluster.
- **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.

For more information on advisors, see [Chapter 20, *Expression-Based Advisor Reference*](#) and [Chapter 21, *GUI-Based Advisor Reference*](#).

Events and Notifications

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 in the **Events** view. 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.

Chapter 3 What's New

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

MySQL Enterprise Monitor Versioning

MySQL Enterprise Monitor 8.0 was reversioned as MySQL Enterprise Monitor 8.0.14 to bring it inline with the MySQL Server and product suite versioning. Versions 8.0.4 to 8.0.13 were skipped.

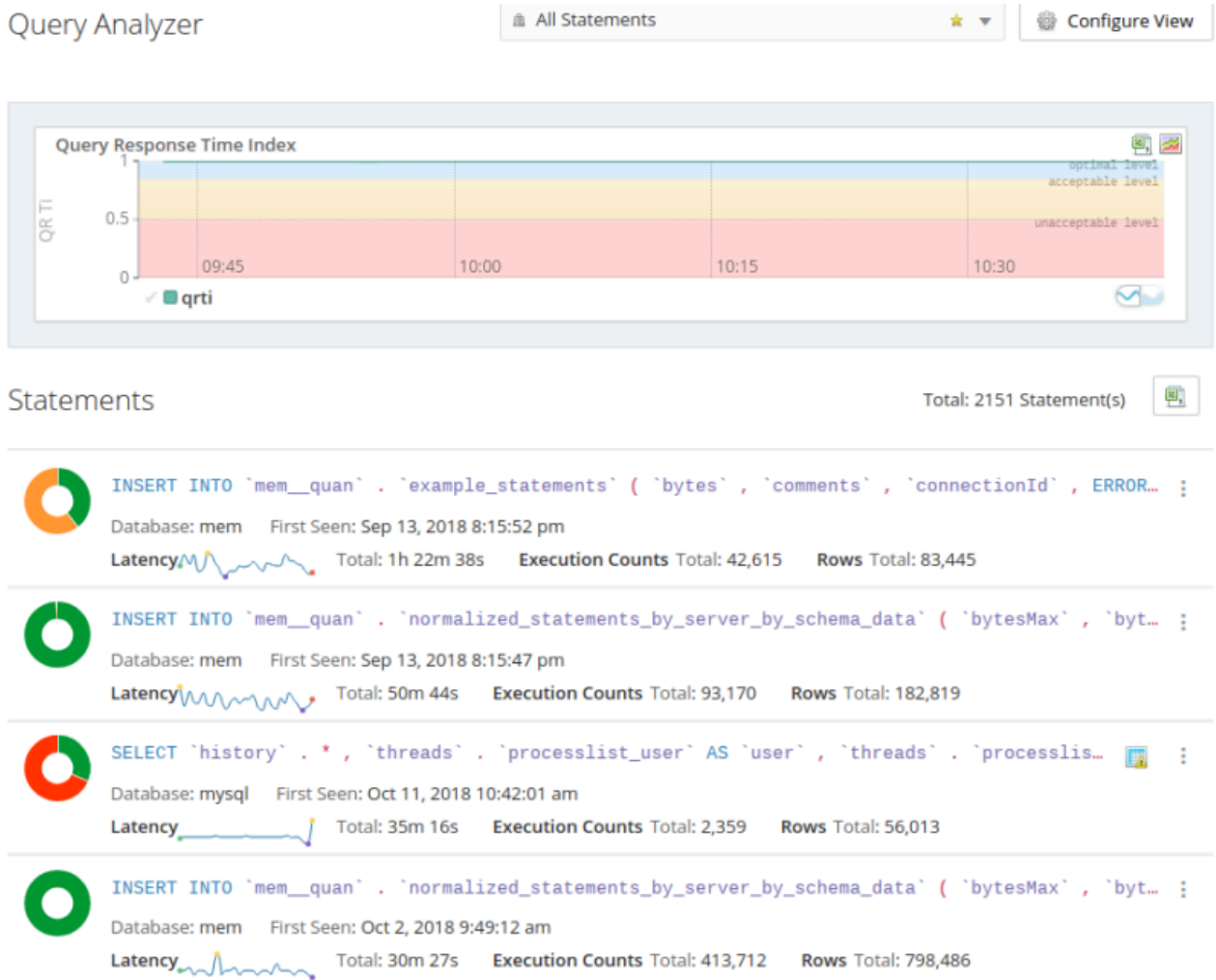
MySQL 8

MySQL Enterprise Monitor now includes MySQL 8. The bundled repository has been upgraded to the latest MySQL server release.

Query Analyzer

The user interface of the Query Analyzer has been completely redesigned. Building upon Oracle's JavaScript Extension Toolkit (JET) framework, this new design improves navigation in the Query Analyzer and introduces advanced configuration options which enable you to customize, sort, and filter the data displayed.

Figure 3.1 Default Query Analyzer View



For more information, see [Section 32.3, "Query Analyzer User Interface"](#).

New Asset Views

Three new asset views are added in this release, **Agents**, **Hosts**, and **NDB Clusters**. These views enable you to examine the configuration and current status of these assets in great detail.







The Agents view configuration view enables you to view all MySQL Enterprise Monitor Agents currently registered with the MySQL Enterprise Service Manager.

The agents are displayed on individual lines, with brief summaries of their current status, number of assets monitored by the agent, and last contact.

Figure 3.2 Agents View

Agents All Agents ★ ⚙️ Configuration View

Select All ✕ Delete Total: 80 Agents

 REACHABLE	Agent01 Version: 8.0.13.1154	1 Monitored Assets	MySQL01:3306 Version: 5.7.24-log	Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:44 PM Data Dir: /var/lib/mysql/	⋮
 REACHABLE	Agent02 Version: 8.0.13.1154	1 Monitored Assets	MySQL02:3306 Version: 5.7.24-ndb-7.6.8-cluster-gpl	Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:34 PM Data Dir: /var/lib/mysql/	⋮
 REACHABLE	Agent03 Version: 8.0.13.1154	0 Monitored Assets	-	-	⋮
 REACHABLE	Agent04 Version: 8.0.13.1154	0 Monitored Assets	-	-	⋮
 REACHABLE	Agent05 Version: 8.0.13.1154	1 Monitored Assets	MySQL05:3306 Version: 5.7.24-log	Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:41 PM Data Dir: /var/lib/mysql/	⋮
 REACHABLE	Agent06 Version: 8.0.13.1154	1 Monitored Assets	MySQL06:3306 Version: 5.7.24-log	Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:29 PM Data Dir: /var/lib/mysql/	⋮

The Agent details view displays all relevant information for that agent. This information includes JVM configuration and status, details on all MySQL connections, processes, and instances, and detailed information on the host's operating system.

Figure 3.3 Agent Details View

List > Agent01

Agent: Agent01

Host Name: Host01
Home Directory: /home/mysql/agent/bin/..
Id: b186b8b0-96a7-e4fb-6a6b-ebc8d6817312
Version: 8.0.13.1154
AGENT REACHABLE

JVM MySQL Connections MySQL Processes MySQL Instances OS

```
Awt Graphicsenv: sun.awt.X11GraphicsEnvironment
Awt Printerjob: sun.print.PSPrinterJob
Class Path: /home/mysql/agent/bin/bootstrap
Class Version: 52.0
Endorsed Dirs: /home/mysql/agent/java/lib/endorsed
Ext Dirs: /home/mysql/agent/java/lib/ext:/usr/java/packages/lib/ext
Home: /home/mysql/agent/java
Io Tmpdir: /tmp
Library Path: /home/mysql/agent/bin/../lib
Runtime Version: 1.8.0_192-b12
Specification Name: Java Platform API Specification
Specification Vendor: Oracle Corporation
Specification Version: 1.8
Vendor: Oracle Corporation
Vendor Url: http://java.oracle.com/
Version: 1.8.0_192
VM Info: mixed mode
VM Name: Java HotSpot(TM) Server VM
VM Specification Name: Java Virtual Machine Specification
VM Specification Vendor: Oracle Corporation
VM Specification Version: 1.8
VM Vendor: Oracle Corporation
VM Version: 25.192-b12
```

The **Hosts** configuration view enables you to view all currently monitored hosts and their configuration.

The hosts are displayed on individual lines, with brief summaries of their current status, number of instances installed, and information on their operating systems and hardware.

Figure 3.4 Hosts View

Hosts		All Hosts	Configuration View	Total: 80 Hosts
	Host01 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores)	Total RAM: 251.77 GiB Total Swap: 4 GiB
	Host02 OS: Oracle Solaris 11 (sparc) Version: 11	MySQL Instances 0	Filesystems: 30 Networks: 2 CPUs: 16 (16 cores)	Total RAM: 32 GiB Total Swap: 8 GiB
	Host03 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores)	Total RAM: 251.77 GiB Total Swap: 4 GiB
	Host04 OS: Oracle Linux 7.2 (x86_64) Version: 7.2	MySQL Instances 0	Filesystems: 96 Networks: 18 CPUs: 88 (88 cores)	Total RAM: 503.78 GiB Total Swap: 4 GiB
	Host05 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores)	Total RAM: 251.77 GiB Total Swap: 4 GiB
	Host06 OS: Oracle Linux 6.5 (x86_64) Version: 6.5	MySQL Instances 1	Filesystems: 15 Networks: 2 CPUs: 2 (2 cores)	Total RAM: 7.55 GiB Total Swap: 4 GiB
	Host07 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores)	Total RAM: 251.77 GiB Total Swap: 4 GiB

The Host detail view displays all relevant information for that host. This information includes filesystem data, and hardware configuration such as RAM, CPUs, and network interfaces.

Figure 3.5 Host Details View

List > {os.Os : Host01}

Host: {os.Os : Host01}

```

Architecture: x86_64
Description: Oracle Linux 7.5
Machine: x86_64
Name: Linux
Patchlevel: unknown
Vendor: Oracle
Vendor Code Name:
Vendor Name: Linux
Vendor Version: 7.5
Version: 4.14.35-1818.0.9.el7uek.x86_64
    
```

Type	System Type	Total Bytes	Used Bytes	Unused Bytes	Available Bytes	Dev Name	Dir Name
none	tmpfs	64 MIB		64 MIB	64 MIB	tmpfs	/proc/keys
local	ext4	365.64 GiB	105.15 GiB	260.49 GiB	241.89 GiB	/dev/sdc1	/home/mysql/agent/etc/agentM
none	tmpfs	125.88 GiB		125.88 GiB	125.88 GiB	tmpfs	/sys/firmware
none	mqueue					mqueue	/dev/mqueue
none	cgroup					cgroup	/sys/fs/cgroup/net_cls,net_prio
none	cgroup					cgroup	/sys/fs/cgroup/cpu,cpuacct
local	ext4	365.64 GiB	105.15 GiB	260.49 GiB	241.89 GiB	/dev/sdc1	/home/mysql/agent/spool
none	cgroup					cgroup	/sys/fs/cgroup/bklcio
local	ext4	365.64 GiB	105.15 GiB	260.49 GiB	241.89 GiB	/dev/sdc1	/etc/resolv.conf
none	tmpfs	64 MIB		64 MIB	64 MIB	tmpfs	/proc/kcore
none	cgroup					cgroup	/sys/fs/cgroup/hugetlb

The **NDB Clusters** configuration view enables you to view all currently monitored clusters, their membership, and configuration.

Figure 3.6 NDB Clusters View

NDB Clusters All Clusters Configuration View

Select All Delete Total: 1 NDB Cluster


 mycluster Replicas: 2 Diskless: OFF	2	4	4	⋮
	Management Nodes	API Nodes	Data Nodes	

Figure 3.7 NDB Clusters Detail View

List > mycluster
 NDB Cluster: mycluster

Diskless Enabled: OFF
 Name: mycluster
 Number Of Replicas: 2

NDB Api Nodes
NDB Data Nodes
NDB Management Nodes

Node Id	Status	Bind Address	Process Name	Arbitration Rank	Auto Reconnect	Backoff Max Time	Sta
mycluster.10/memcached	Unknown	172.18.0.10	memcached	2	ON	1500 (25m)	0
mycluster.7/mysqld	up	172.18.0.8	mysqld	2	ON	1500 (25m)	0
mycluster.8/mysqld	up	172.18.0.9	mysqld	2	ON	1500 (25m)	0
mycluster.9/mysqld	up	172.18.0.10	mysqld	2	ON	1500 (25m)	0

For more information, see [Chapter 17, Asset Management](#)

Deleting Assets from the Inventory

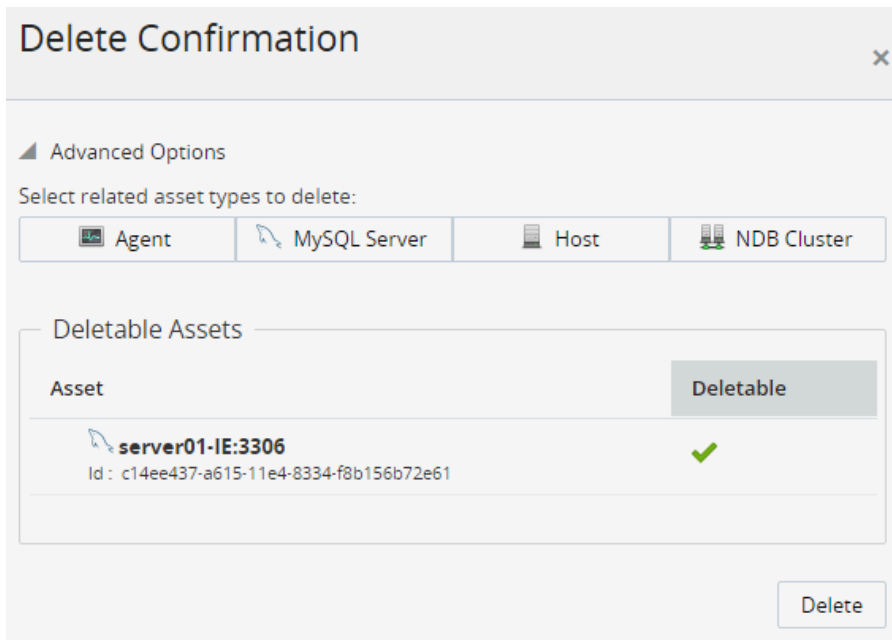
It is now possible to delete assets from the MySQL Enterprise Monitor inventory from the user interface. You can delete individual assets or entire groups of related assets.



Important

Deleting assets from the inventory does not delete or uninstall assets from their installed locations. The delete process removes all record of an asset or group of assets from the internal record of MySQL Enterprise Service Manager, only. This enables you to remove entries for clusters, topologies, agents, hosts, and instances which were monitored, but are no longer in use.

Figure 3.8 Delete Confirmation for MySQL Instance



It is also possible to multi-select assets for deletion.

For more information, see [Section 17.4, "Deleting Assets"](#)

Part II Installing MySQL Enterprise Monitor

Table of Contents

4 Prerequisites	25
4.1 Installer Files	25
4.2 Prerequisites	25
4.2.1 System Requirements	25
4.2.2 Supported Platforms	26
4.2.3 Requirements for Monitoring	27
4.2.4 MySQL Enterprise Monitor Repository	28
4.3 Credentials Required for Installation	30
4.3.1 Existing Users	30
4.3.2 Users Created During Installation	31
4.3.3 Users Created on First Log-in	31
4.4 Supported Browsers	31
5 Service Manager Installation	33
5.1 MySQL Enterprise Monitor Installation Types	33
5.2 MySQL Enterprise Service Manager Graphical Installation Wizard	34
5.3 Text-Based Installation	36
5.4 Starting/Stopping MySQL Enterprise Monitor Services	37
5.5 MySQL Enterprise Service Manager Configuration Settings	38
6 Monitor Agent Installation	41
6.1 General Agent Related Notes	41
6.2 Creating MySQL User Accounts for the Monitor Agent	42
6.3 Java Considerations on UNIX-like Platforms	45
6.4 Monitoring Agent Graphical Installation Wizard	45
6.5 Starting/Stopping MySQL Enterprise Monitor Agent	48
6.5.1 Starting/Stopping the Agent on Windows	48
6.5.2 Starting/Stopping the Agent on macOS	49
6.5.3 Starting/Stopping the Agent on Unix	50
6.5.4 sql_mode	51
6.6 Monitoring Multiple MySQL Servers	51
6.7 Configuring an Agent to Monitor a Remote MySQL Server	51
6.8 Monitoring Outside the Firewall with an SSH Tunnel	52
6.9 HTTP Connection Timeout	52
6.10 Troubleshooting the Agent	52
6.11 Agent Backlog	54
7 Upgrading MySQL Enterprise Monitor Installations	55
7.1 General considerations when upgrading MySQL Enterprise Monitor	55
7.2 Upgrading MySQL Enterprise Monitor	55
7.3 Restoring from Backup	57
8 Post-installation Considerations	59
8.1 General Considerations	59
8.2 Installing SSL Certificates	60
8.3 Backup the Repository	63
8.4 Changing an SSH Host Key	64
9 Unattended Installation Reference	67
9.1 Unattended Installation	67
9.1.1 Performing an Unattended Installation	67
9.1.2 MySQL Enterprise Service Manager Options	68
9.1.3 MySQL Enterprise Monitor Agent Options	74
10 Performance Tuning MySQL Enterprise Monitor	83
10.1 Tuning Memory	83
10.2 Tuning CPU	84

10.3 Tuning Apache Tomcat Threads	85
10.4 Tuning Agent Memory Requirements	85
11 Configuration Utilities	87
11.1 Service Manager Configuration Utilities	87
11.2 Agent Configuration Utility	89
12 Uninstalling MySQL Enterprise Monitor	93
12.1 Windows Platforms	93
12.2 macOS Platforms	94
12.3 Unattended Uninstallations	95

Chapter 4 Prerequisites

This chapter describes the process of installing 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 MySQL Enterprise Service Manager installation. See [Chapter 5, Service Manager Installation](#).
- Optionally (but recommended), one or more MySQL Enterprise Monitor Agents, one on each host being monitored. Install MySQL Enterprise Service Manager first, because the Agent installation asks for credentials and network settings that you choose as you install 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 MySQL Enterprise Service Manager. An agent can monitor locally or remotely.



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 parse the results and display the data in the browser-based MySQL Enterprise Monitor User Interface.

After installing and starting the Service Manager and Agents, configure the settings as described in [Section 5.5, “MySQL Enterprise Service Manager Configuration Settings”](#).

4.1 Installer Files

MySQL Enterprise Monitor files include:

- MySQL Enterprise Service Manager: 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, MySQL Enterprise Monitor Agent is installed on the same machine as a monitored MySQL instance. 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`.

4.2 Prerequisites

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

4.2.1 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 MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent on each platform.

Table 4.1 Required Disk Space

Platform	Service Manager Minimum Disk Space	Agent Minimum Disk Space
Linux x86 64-bit	1.3 GB	800 MB
macOS	1.2 GB	700 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.

4.2.2 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.
- On Linux platforms, ensure you have both the `libaio` and `libnuma` libraries installed. The installer checks for both.
- MySQL Enterprise Service Manager installation generates a self-signed certificate during the installation process. This certificate generation requires a valid, resolvable hostname. It is not possible to install MySQL Enterprise Service Manager on a host which does not have a valid hostname.

MySQL Requirements

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

- 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 MySQL Enterprise Service Manager installation, see [Section 4.2.4, “MySQL Enterprise Monitor Repository”](#) for more information.
- As of MySQL Enterprise Monitor 8.0.20, the MySQL Enterprise Service Manager installer, on Microsoft Windows, checks for the presence of the Microsoft Visual C++ 2015-2019 Redistributable (x64) - 14.25.2850508™ or later and if it is not present, installs it.

MySQL Enterprise Service Manager installation on Microsoft Windows can not proceed without this redistributable.



Important

If you do not intend to use the bundled MySQL Server, but intend to use an existing, external repository, you must manually download and install this redistributable.

- 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.

4.2.3 Requirements for Monitoring

This section describes the requirements for monitoring MySQL Server.

- The Monitoring Agent officially supports MySQL Server 8.0 as per [Supported Platforms: MySQL Enterprise Monitor](#). However, MySQL Enterprise Monitor makes an effort to monitor any GA version of MySQL Server from version 5.5 onwards.



Important

It is not possible to monitor pre-GA versions of MySQL.

- 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.

- If you intend to monitor MySQL NDB Cluster, you must ensure the following when configuring your cluster:
 - NDB Cluster version is 7.5.7, or higher
 - Edit the `[System]` section of your cluster configuration file to include the cluster name. MySQL Enterprise Monitor Agent reads this value and uses it to uniquely identify your cluster. For example:

```
[System]
name=cluster-name
```

- It is strongly recommended to install MySQL Enterprise Monitor Agent on all Management, Data, and SQL nodes for optimal monitoring. If this is not possible, monitoring the SQL nodes provides useful information about the entire cluster.
- Always include the `--ndb-connectstring` and `--ndb-nodeid` flags in the corresponding command to facilitate topology discovery.
- The Management node was started with the `--config-file` flag, and ensure the owner of MySQL Enterprise Monitor Agent process has permissions to read the configuration file.
- If your host has multiple addresses, include the `--bind-address` flag on the `ndb` commands, matching the `HostName` variable defined in the configuration file.
- Do not use `localhost` or any loopback address in your configuration or commands. Always use a unique address.

4.2.4 MySQL Enterprise Monitor Repository

MySQL 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 installed on the same machine as MySQL Enterprise Service Manager, or on a remote machine.



Important

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

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

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

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

If you intend to use an alternative MySQL instance for your MySQL Enterprise Service Manager repository, the repository instance **must** be present before starting 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 5.7.24, or higher, or 8.0.13, or higher.



Note

It is not possible to use any MySQL 5.7 version preceding 5.7.24, or any MySQL 8.0 version preceding 8.0.13, as MySQL Enterprise Monitor repository. If you attempt to configure MySQL Enterprise Service Manager to use an earlier version, the installer displays an error and the installation cannot proceed.

- The InnoDB storage engine is available. You must use InnoDB as the default storage engine.
- SSL is enabled.

You must ensure the following in the MySQL Server configuration:

- Query Cache must not be enabled.
- Set `default-storage-engine=InnoDB`.
- Set `innodb_file_per_table=1`.
- Set `innodb_file_format=Barracuda`. This is not required if you are configuring MySQL 8 as the external repository.
- On Linux/Unix hosts, ensure `innodb_flush_method=O_DIRECT`.
- It is recommended to set `innodb_log_file_size=2048M`.
- `performance-schema-consumer-events-statements-history-long=ON` must be added to the repository's configuration file.
- Define a Service Manager user to enable MySQL Enterprise Service Manager to connect to, and modify, the repository. This user must have the following privileges:

- All privileges on `mem%.*` tables

For example:

```
GRANT ALL PRIVILEGES ON `mem%`.* TO 'service_manager'@'127.0.0.1';
```

- `REPLICATION CLIENT`, `SUPER`, `PROCESS`, and `SELECT` on all databases in the repository.

For example:

```
GRANT REPLICATION CLIENT, SUPER, PROCESS, SELECT ON *.* TO 'service_manager'@'127.0.0.1';
```

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



Important

If you are using a MySQL Server repository other than the installation bundled with MySQL Enterprise Service Manager installation, the installer checks your MySQL Server for the following, minimum requirements:

- MySQL Server version: versions older than MySQL 5.7.9 are not supported.

If you intend to use MySQL 8, it is not possible to use any version preceding MySQL 8.0.13. That is, it is not possible to use 8.0.0 to 8.0.4 as repository.

- SSL: SSL must be enabled on the MySQL Server.
- For MySQL 5.7, `innodb_file_format: innodb_file_format` must be set to `barracuda`. This variable is deprecated in MySQL 8 and is not checked.
- `innodb_file_per_table: innodb_file_per_table` must be enabled.

If any checks fail, the installer displays an error and cannot proceed until the configuration is corrected.

The upgrade process performs the same checks.

To check your repository for compatibility, before upgrading, use MySQL Enterprise Service Manager configuration utility. For more information, see [Section 11.1, "Service Manager Configuration Utilities"](#).

4.3 Credentials Required for Installation

Before installing MySQL Enterprise Monitor, gather credentials (a root user ID and password) for all the MySQL servers you plan to monitor. MySQL Enterprise Monitor Agent installation requires a dedicated user in each monitored MySQL server, and optional limited and general users 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 plug-in](#). Currently, MySQL Enterprise Monitor does not support authentication through the [Windows Native Authentication plug-in](#).

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

4.3.1 Existing Users

The **MySQL user**: For Agents to report the status of a MySQL server, they connect to a MySQL user with privileges to read required 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 6.2, "Creating MySQL User Accounts for the Monitor Agent"](#).

4.3.2 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. Depending on platform, this file is located in the following directories:

- Windows: `C:\Program Files\MySQL\Enterprise\Monitor`
- Unix: `/opt/mysql/enterprise/monitor`
- macOS: `/Applications/mysql/enterprise/monitor`

4.3.3 Users Created on First Log-in

The **Manager user**: This user is the administrator of MySQL Enterprise Service Manager. Specify a user name and password.

The **Agent user**: MySQL Enterprise Monitor Agent broadcasts the status of the assets it is monitoring and requires access to MySQL Enterprise Service Manager. Define the user name and password for this user. These credentials must also be provided for the agent installation process.



Note

The Monitor Agent communicates both with MySQL Enterprise Service Manager, and with the MySQL server it is monitoring. For a description of the agent as a MySQL user, see [Section 4.3.1, “Existing Users”](#).

4.4 Supported Browsers

The following browser versions are recommended for use with MySQL Enterprise Monitor User Interface:

- Microsoft Internet Explorer 11 and Microsoft Edge.
- 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 5 Service Manager Installation

This chapter describes the installation of MySQL Enterprise Service Manager.

MySQL Enterprise Service Manager installer installs the following components:

- Apache Tomcat: mandatory component. Servlet container and web server which hosts 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. For more information, see [Section 4.2.4, “MySQL Enterprise Monitor Repository”](#).

5.1 MySQL Enterprise Monitor Installation Types

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 5.1 Installation Parameters

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

Parameter	Small	Medium	Large
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`.

5.2 MySQL Enterprise Service Manager Graphical Installation Wizard

This section describes how to install 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 5.1, “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.
- **Tomcat SSL Port:** Default value is 18443. This port is mandatory for communication with 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 must run as. 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 the following: .

```
GRANT ALL PRIVILEGES ON *.* TO 'SM_UserName'@'localhost'  
WITH GRANT OPTION;
```

where `SM_Username` is the user name of MySQL Enterprise Service Manager user.

- **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 4.2.4, “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 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 macOS 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**.



Important

If you are attempting to use a version of MySQL older than MySQL Server 5.7.9 as an external repository, an error is displayed and the installation fails. MySQL Server 5.7.9 is the minimum version and MySQL Server 5.7.18 is the recommended version.

For more information, see [Section 4.2.4, “MySQL Enterprise Monitor Repository”](#).

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.

5.3 Text-Based Installation

The steps and options of the text-based installation are identical to those described in [Section 5.2, “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:

```
$> ./mysqlmonitor-4.0.3.5191-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 5.2, “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 5.5, “MySQL Enterprise Service Manager Configuration Settings”](#).

5.4 Starting/Stopping MySQL Enterprise Monitor Services

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

The following services are installed by MySQL Enterprise Service Manager:

- MySQL Server
- Tomcat Server

Access 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:

```
$> ./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:

```
$> ./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.

```
$> ./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 5.5, “MySQL Enterprise Service Manager Configuration Settings”](#).

Starting/Stopping MySQL Enterprise Monitor Services on Windows

You can stop or start 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.

5.5 MySQL Enterprise Service Manager Configuration Settings

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** check box is checked, which it is by default).

Otherwise, you can view 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 6 Monitor Agent Installation

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



Important

It is strongly recommended that MySQL Enterprise Monitor Agent 8.0 are installed for use with the MySQL Enterprise Service Manager 8.0.

6.1 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 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 MySQL Enterprise Monitor User Interface. From there, you can add the appropriate configuration information.
- To detect a local connection in IPv6, the Agent requires forward resolution configured 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.

- You can assign a monitored MySQL instance to a group via the Agent installer, which is displayed in MySQL Enterprise Service Manager.
- The old Agent's configuration files (`mysql-monitor-agent.ini` and `agent-instance.ini`) no longer exist. Use `custom.xml` instead.
- Passwords are now stored in an encrypted format, so you can no longer recover passwords by looking in the configuration files.
- The Service Manager now bundles an Agent, which monitors the host on which it is installed, scans for all MySQL instances on the host, and also monitors the Service Manager repository database.



Note

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

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

6.2 Creating MySQL User Accounts for the Monitor Agent

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` privilege. The `SUPER` privilege is required to run certain statements such as `SHOW MASTER LOGS` or `SHOW ENGINE INNODB STATUS`, depending on the version that is being monitored.



Note

If monitoring MySQL Server 5.5, you must also grant `CREATE` and `INSERT` privileges on the schema the inventory table is created on. The inventory table stores unique identifiers for the MySQL instance, and is created in the `mysql` schema by default.

The inventory table is not used in MySQL Server 5.6, or higher.

If you intend to automatically create the less-privileged users, General and Limited, you must also grant the Admin user `CREATE USER`. You must also grant `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`, `INSERT`, `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, grant at least the `EXECUTE`, `PROCESS`, `REPLICATION CLIENT`, `SELECT`, and `SHOW DATABASES` privileges globally, and `UPDATE` on the `performance_schema.threads` table. If you intend to use `EXPLAIN` on views, you must also grant `SHOW VIEW`.



Important

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

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

Examples of these types of statements include getting database metadata from `INFORMATION_SCHEMA` tables, 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 `EXECUTE`, `SELECT` and `SHOW DATABASES` privileges globally, and `UPDATE` on the `performance_schema.threads` table. If you intend to use `EXPLAIN` on views, you must also grant `SHOW VIEW`.

Creating the Admin User

If you do not want to supply the root user information to the installer, create a user manually within your MySQL server and provide these credentials as the agent user/password combination during installation. The privileges required for this user account vary depending on the information you gather using the MySQL Enterprise Monitor Agent.

- `SHOW DATABASES`: MySQL Enterprise Monitor Agent can gather inventory about the monitored MySQL server.
- `REPLICATION CLIENT`: 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`: MySQL Enterprise Monitor Agent can collect statistics for table objects.
- `SUPER`: 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, if monitoring MySQL 5.5 or earlier, to temporarily switch off replication when populating the `mysql.inventory` table used to identify the MySQL instance. The `mysql.inventory` table is not created on MySQL 5.6, or higher.



Note

`SELECT` is required to read data on all objects from `INFORMATION_SCHEMA`. If it is not granted, it is not possible for MySQL Enterprise Monitor to perform object monitoring or statistics.

- `EXECUTE`: required to execute helper functions.
- `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`: If you are monitoring a MySQL 5.6, or higher, this privilege is not required. Required to create the UUID used by the agent.
- `CREATE`: If you are monitoring a MySQL 5.6, or higher, this privilege is not required. 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.

- `DROP` on the `ndb_objects` table, a requirement for MySQL Enterprise Monitor Agent 8.0.22 or higher.

For example, the following `GRANT` statement gives the agent the required rights:

```
GRANT SELECT, CREATE USER, REPLICATION CLIENT, SHOW DATABASES, SUPER, DROP, PROCESS
ON *.*
TO 'agent_user'@'localhost';
```



Note

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

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

```
GRANT CREATE, DROP, INSERT
ON mysql.*
TO 'agent_user'@'localhost';
```

If the master, or slave, runs a version of MySQL earlier than 5.6, you must grant the `SELECT` privilege on the `mysql.inventory` table for each user with replication privileges on the corresponding replication master. This enables MySQL Enterprise Monitor Agent to read the replication master UUID. For example:

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



Note

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



Important

If running MySQL 5.6, or later, it is not necessary to grant select on `mysql.inventory` because the `master_uuid` is used from the slave status. Therefore, there is no need to start and stop the agent, nor to set `SQL_LOG_BIN` to 0.

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** check box, 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 6.7, “Configuring an Agent to Monitor a Remote MySQL Server”](#).

6.3 Java Considerations on UNIX-like Platforms

MySQL Enterprise Monitor Agent installers and updaters for UNIX-like 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.

6.4 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

Although 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. If you attempt to install to a directory on which the `mysql` user cannot act, an error is displayed explaining how to correct the situation.

You can also install the Monitor Agent in `unattended` mode. For more information on unattended installation, see [Section 9.1, “Unattended Installation”](#).



Note

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



Note

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 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 monitors.
 - **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.

6.5 Starting/Stopping MySQL Enterprise Monitor Agent

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. MySQL Enterprise Service Manager provides a warning if an agent and the MySQL server that it monitors are unavailable.

6.5.1 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 6.10, “Troubleshooting the Agent”](#).

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

```
$> sc start MySQLEnterpriseMonitorAgent
```

or:

```
$> 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:

```
$> sc stop MySQLEnterpriseMonitorAgent
```

or:

```
$> net stop MySQLEnterpriseMonitorAgent
```



Note

[MySQLEnterpriseMonitorAgent](#) is the default name of the Monitor Agent service.

6.5.2 Starting/Stopping the Agent on macOS

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 start or stop the Agent service using the following commands.

To start the Agent:

```
$> sudo launchctl load /Library/LaunchDaemons/mysql.agent.plist
```

To stop the Agent:

```
$> sudo launchctl unload /Library/LaunchDaemons/mysql.agent.plist
```

Using init

Alternatively, an `init.d` script to start the Agent on macOS 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:

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

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

```
$> ./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:

```
$> 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:

```
$> ./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:

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

For further information on troubleshooting the agent, see [Section 6.10, “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:

```
$> ./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 6.10, “Troubleshooting the Agent”](#).

6.5.3 Starting/Stopping the Agent on Unix

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

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

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

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

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

```
$> ./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:

```
$> kill -TERM PID
```

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

```
$> ./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:

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

For further information on troubleshooting the agent, see [Section 6.10, “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:

```
$> ./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 6.10, “Troubleshooting the Agent”](#).

6.5.4 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 MySQL Enterprise Service Manager also sets `sql_mode=STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION` on the repository.

6.6 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 6.2, “Creating MySQL User Accounts for the Monitor Agent”](#).

Typically, an Agent scans a host and reports unmonitored MySQL instances to MySQL Enterprise Monitor User Interface. For more information, see [Section 2.2, “MySQL Enterprise Monitor Agent”](#). For information about how to change the status of a MySQL instance from unmonitored to monitored, see [Chapter 16, Manage Instances](#).

6.7 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 6, 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 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 6.2, “Creating MySQL User Accounts for the Monitor Agent”](#).

The Agent also needs to be able to log in to 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, where a loss of connection means loss of

monitoring data, whereas an Agent installed on the same server as the MySQL instance continues monitoring and transmits the backlog of data to the MySQL Enterprise Service Manager upon reconnection.

- For replication autodiscovery, do not use remote monitoring for replicas or sources. The Agent must be installed on the same machine as the server you are monitoring for discovery to work properly. For more information, see [Chapter 28, Replication Dashboard](#).

6.8 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:

```
$> 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 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 6.5.1, “Starting/Stopping the Agent on Windows”](#).
- Under Unix see, [Section 6.5.3, “Starting/Stopping the Agent on Unix”](#).
- Under macOS see, [Section 6.5.2, “Starting/Stopping the Agent on macOS”](#).

6.9 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.

6.10 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 6.5.1, “Starting/Stopping the Agent on Windows”](#)
- Unix: [Section 6.5.3, “Starting/Stopping the Agent on Unix”](#)
- macOS: [Section 6.5.2, “Starting/Stopping the Agent on macOS”](#)

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 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, you may be 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 is displayed, and you are unable to connect, the host name or port number might be specified incorrectly. Confirm the values you entered against those described as the `Application hostname and port:` in the `configuration_report.txt`. Failure to connect could also indicate that the port is blocked on the machine hosting MySQL Enterprise Service Manager.
- To check if a blocked port is the problem, check your firewall's configuration for that port number. Open 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 6.8, “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 6.7, “Configuring an Agent to Monitor a Remote MySQL Server”](#).
- On MySQL 5.5, or earlier, 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.

6.11 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 7 Upgrading MySQL Enterprise Monitor Installations

This chapter describes the upgrade process for MySQL Enterprise Monitor 8.0.



Important

It is not possible to upgrade any version older than MySQL Enterprise Monitor 3.4.0 directly to 8.0. To upgrade an older installation, you must first upgrade to 3.4.x, then upgrade to 8.0.

The upgrade installer generates an error if you attempt to upgrade an older version.

7.1 General considerations when upgrading MySQL Enterprise Monitor

You cannot use the update installers to update a 32-bit Linux installation to a 64-bit version using an update installer. You must perform a fresh installation.

Customizations to `setenv.sh` are lost, as this file is replaced and optimized 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 file names 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.



Important

MySQL Enterprise Service Manager is bundled with the latest MySQL server. If you are using the bundled MySQL server for your repository (default installation), the upgrade process upgrades your repository to the latest version bundled. If you are using another MySQL Server as your repository, you must upgrade it manually before installing MySQL Enterprise Service Manager.

If you attempt to configure a repository which does not meet the version prerequisites, the upgrader displays an error message, and the installation cannot continue.

7.2 Upgrading MySQL Enterprise Monitor

You must upgrade your MySQL Enterprise Service Manager installation first, and your agents after the MySQL Enterprise Service Manager installation completes successfully.

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 *mysqlenterprise-8.0.14-windows-update-installer.exe* updates MySQL Enterprise Service Manager on Microsoft Windows to version 8.0.14.

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 9.1.2, “MySQL Enterprise Service Manager Options”](#).

Service Manager Upgrade Process



Important

If using an external repository, you must upgrade it to at least MySQL Server 8.0.13 before upgrading your MySQL Enterprise Service Manager. 8.0.13 is recommended.

To upgrade MySQL Enterprise Service Manager, do the following:

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



Note

If you are using LDAP and have imported your own SSL certificates into the keystore of the Java VM bundled with your MySQL Enterprise Monitor, you must extract them from the backup listed below and manually re-import them, as this upgrade included a new Java VM.

Agent Upgrade Process

To upgrade MySQL Enterprise Monitor Agent, do the following:

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** check box, 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 check box.

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

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

The installation is started and completes.

7.3 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 8 Post-installation Considerations

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

8.1 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 connections, select **Instances** from the **Configuration** section of the navigation 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 want to monitor. See [Section 16.2, “Monitored Instance Details”](#) for more information on creating connections in the User Interface.
- Select **Groups** from the **Configuration** section of the navigation menu to collect your MySQL instances into Groups.

2. **Users, Roles, and Access Control (ACLs):** Before creating your user accounts, see [Chapter 23, Access Control](#) and [Chapter 24, Access Control - Best Practices](#).

Do you want to provide open access to all monitored resources for all Users? Or 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 18, 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 navigation menu. Complete, and test, the SMTP, or SNMP, configuration. See [Chapter 22, Event Handlers](#) for more information.
4. **Overview:** Select **Overview** from the navigation menu. Set the defaults for the groups you want to view, the time range, and graphs to display. See [Chapter 25, Overview](#) for more information.
5. **Replication:** If you are using MySQL Replication, select **Replication** a from the navigation menu and select a group to view its configuration, status and replication error details. Select **Topology** a from the navigation menu to see the topology of the replication. See [Chapter 28, Replication Dashboard](#) for more information.
6. **Advisors:** You can accept the defaults defined, or select **Advisors** from the **Configuration** section of the navigation menu and customize the threshold for groups, or individual MySQL Instances. For more information, see [Chapter 19, Advisors](#).
7. **SQL Performance Tuning:** If you are monitoring instances of MySQL running version 5.6.14 or later, rich SQL performance tuning data is available in the **Query Analyzer**.

8. **I/O and Lock Contention:** If you are using MySQL 5.6 or later consider deploying the sys schema, and making use of the **Database File I/O and Lock Waits** reports from the Reports & Graphs menu. These 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 30.2, “Database File I/O and Lock Waits”](#) 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 7, *Upgrading MySQL Enterprise Monitor Installations*](#).
- **Overview:** If you have not already done so, select **Overview** from the navigation menu and familiarize yourself with the updated design. Then set defaults for which Group you want to view, the graph time range, and the set and order of Graphs to display. See [Chapter 18, *Managing Groups of Instances*](#).

8.2 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 all certificate-related activities, it is recommended to use the MySQL Enterprise Monitor Configuration Utility. See [Chapter 11, *Configuration Utilities*](#) for more information.

To generate the certificate and add it to the default keystore, run the following command with the MySQL Enterprise Monitor Configuration Utility:

```
config --renew
```

This regenerates the self-signed certificate. If the certificate is not self-signed, an error is displayed.

MySQL Enterprise Service Manager SSL Import

To install an SSL certificate for the MySQL Enterprise Service Manager, use the MySQL Enterprise Monitor Configuration Utility to import the certificate into the keystore.

To import your certificate, run the following command:

```
config --import-certificate=/path/to/client.crt
```

To import an existing certificate and key, run the following command:

```
config --import-certificate=/path/to/client.crt --import-key=/path/to/client.key
```

Restart the service manager. For more information about stopping and starting the service manager, see the instructions for [Unix/macOS](#) 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 8.1 Agent SSL Configuration Options

Parameter	Description
<code>ssl-verify-hostnames</code>	<p>Values: True or False (default).</p> <p>SSL connection between service manager and agent.</p> <p>Verify that the hostname of the service manager that the agent is connected to matches what is in the SSL certificate.</p>
<code>ssl-allow-self-signed-certs</code>	<p>Values: True (default) or False.</p> <p>SSL connection between agent and MySQL instance.</p> <p>If set to <code>true</code> self-signed certificates are permitted. If set to <code>false</code>, self-signed certificates are not permitted.</p>
<code>ssl-ca-keystore-path</code>	<p>Values: String</p> <p>SSL connection between agent and MySQL instance.</p> <p>Path to keystore with CA cert(s), if <code>ssl-allow-self-signed-certs</code> is true. This path must be defined as a URL. For example:</p> <pre>file:///Applications/mysql/enterprise/agent/etc/mykeystore</pre>
<code>ssl-ca-keystore-password</code>	<p>Values: String</p> <p>SSL connection between agent and MySQL instance.</p> <p>Password for the CA keystore, if <code>ssl-allow-self-signed-certs</code> is true.</p>

An example `bootstrap.properties` SSL certification section:

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

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

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

The tool responds with the certificate details. For example:

```
Enter keystore password: (the keystore requires 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.



Note

The connection between the agent and service manager is also an SSL connection.

LDAP SSL Configuration

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



Important

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

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

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

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

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


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

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

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

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

```
mysql/enterprise/monitor/mysqlmonitorctl.sh restart
```

8.3 Backup the Repository

The following is an example of how to use `mysqldump` to export all databases in the repository instance to a dump file named `mem.dump`:



Note

This example assumes you are using the MySQL instance solely as the repository for MySQL Enterprise Service Manager and not for any other purpose. This is the recommended implementation.

```
$> mysqldump --single-transaction -u service_manager -pPassword  
-P13306 -h127.0.0.1 --all-databases > mem.dump
```

The above command creates a file, `mem.dump`, containing all MySQL Enterprise Service Manager data.

To restore the dump file, run the following `mysql` command on a clean instance:

```
$> mysql -u <user> -p -P13306 -h127.0.0.1 < mem.dump
```

You should also backup the following files:

- `apache-tomcat/conf/Keystore`
- `apache-tomcat/conf/server.xml`
- `java/lib/security/cacerts`
- `apache-tomcat/webapps/ROOT/WEB-INF/config.properties`
- `apache-tomcat/webapps/ROOT/WEB-INF/configArea/mem.keystore`

You can also use MySQL Enterprise Backup to manage your repository backup process. For more information, see the [MySQL Enterprise Backup](#) documentation.



Note

It is also possible to shut down MySQL Enterprise Service Manager, copy the entire data directory to another location, and restart. To restore the copy, simply shut down MySQL Enterprise Service Manager and overwrite the data directory with the backup, and restart. This is the same process used by MySQL Enterprise Service Manager upgrade installer.

8.4 Changing an SSH Host Key

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

The following steps must be performed:

- Generate a new SSH key for the monitored host.
- Edit the monitoring agent's configuration.

On UNIX, Linux and macOS 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. Identify the duplicate SSH keys.

On each monitored host, run the following commands to print out the SSH public key fingerprints:

```
ssh-keygen -l -f /etc/ssh/ssh_host_key.pub
ssh-keygen -l -f /etc/ssh/ssh_host_dsa_key.pub
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

and compare the fingerprints between each server. Remove the existing key and regenerate the key for each server with a duplicate key using the following instructions.

2. On the monitored host, delete the existing, duplicate SSH key. For example:

```
sudo rm -f /etc/ssh/ssh_host_*
```

3. On the monitored host, regenerate the SSH keys. This operation requires you to add each of these hosts to your SSH keyring the next time you connect. For example:

```
sudo ssh-keygen -f /etc/ssh/ssh_host_key -N '' -t rsa1
sudo ssh-keygen -f /etc/ssh/ssh_host_rsa_key -N '' -t rsa
sudo ssh-keygen -f /etc/ssh/ssh_host_dsa_key -N '' -t dsa
```

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.

4. Restart your `sshd` server.

- 5.



Note

This step is required on MySQL 5.5 instances, only. It is not required on any version of MySQL 5.6, or higher.

On the monitored MySQL instance, edit the `hostid` value:

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

6. Restart the monitoring agent.

Chapter 9 Unattended Installation Reference

9.1 Unattended Installation

This section explains how to automate the install and upgrade procedures for 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.

9.1.1 Performing an Unattended Installation

The basic process for performing an unattended installation is the same for both 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 9.1.2, “MySQL Enterprise Service Manager Options”](#). For information on the options for MySQL Enterprise Monitor Agent, see [Section 9.1.3, “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.

The following example shows how to install MySQL Enterprise Monitor Agent by command-line:

```
$> 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 containing the installation parameters. 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 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:

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

On macOS, locate the `installerbuilder.sh` within the installation package directory. For example:

```
$> ./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 MySQL Enterprise Monitor Agent installer and the corresponding agent options.

As a minimum for 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.

9.1.2 MySQL Enterprise Service Manager Options

The following options let you customize the installation process for MySQL Enterprise Service Manager. 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 4.2.4, “MySQL Enterprise Monitor Repository”](#).

Table 9.1 MySQL Enterprise Service Manager Installer Options

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

Option Name	Description
<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>--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>--tomcatsslport</code>	SSL TCP/IP port for the Tomcat component
<code>--unattendedmodeui</code>	Unattended mode user interface
<code>--version</code>	Display the product information

- `--help`

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

Display the list of valid installer options.

- `--version`

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

Display product and version information.

- `--backupdir`

Command-Line Format	<code>--backupdir</code>
Type	String

Upgrade only. The backup directory.

- `--createDataBackup`

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

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

- `--optionfile`

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

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

- `--mode`

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

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

- `--debuglevel`

Command-Line Format	<code>--debuglevel</code>
Type	Numeric
Default Value	2
Minimum Value	0
Maximum Value	4

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

- `--installer-language`

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

The installer language.

- `--installdir`

Command-Line Format	<code>--installdir</code>
Type	String
Default Value (OS X)	<code>/Applications/mysql/enterprise/monitor/</code>
Default Value (Unix)	<code>/opt/mysql/enterprise/monitor/</code>
Default Value (Windows)	<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>
Type	String
Default Value	<code>medium</code>
Valid Values	<code>small</code> (5 to 10 MySQL Servers monitored from a laptop or low-end server with no more than 4GB of RAM.) <code>medium</code> (Up to 100 MySQL Servers monitored from a medium-sized, but shared, server with 4 to 8GB of RAM.) <code>large</code> (More than 100 MySQL Servers monitored from a high-end, dedicated server, with more than 8GB RAM.)

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

- `--tomcatport`

Command-Line Format	<code>--tomcatport</code>
Type	Numeric

Default Value	18080
---------------	-------

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

- `--tomcatsslport`

Command-Line Format	<code>--tomcatsslport</code>
Type	Numeric
Default Value	18443

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

- `--mysql-identity-source`

Command-Line Format	<code>--mysql-identity-source</code>
Type	String
Default Value	<code>default</code>
Valid Values	<code>default</code> (Default) <code>host_plus_datadir</code> (<code>host_plus_datadir</code>)

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.

- `--adminuser`

Command-Line Format	<code>--adminuser</code>
Type	String

Default Value	<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>
Type	String
Default Value	<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>
Type	String

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

- `--mysql_installation_type`

Command-Line Format	<code>--mysql_installation_type</code>
Type	String
Default Value	<code>bundled</code>
Valid Values	<code>bundled</code> (Use the bundled MySQL server) <code>existing</code> (Use an existing (user supplied) MySQL server)

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

- `--dbport`

Command-Line Format	<code>--dbport</code>
Type	Numeric
Default Value	<code>13306</code>

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

- `--dbhost`

Command-Line Format	<code>--dbhost</code>
Type	String
Default Value	<code>127.0.0.1</code>

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

- `--dbname`

Command-Line Format	<code>--dbname</code>
Type	String
Default Value	<code>mem</code>

The name of MySQL Enterprise Service Manager repository.

- `--forceRestart`

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

Force a restart of MySQL Enterprise Service Manager services.

9.1.3 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 9.2 MySQL Enterprise Monitor Agent Installer Options

Option Name	Description
<code>--agent_installtype</code>	Installation type for the agent, which can be database or standalone.
<code>--agent_autocreate</code>	Create an account on the monitored MySQL server to be used by the agent
<code>--agentpassword</code>	Password of the agent user for connecting to the monitored MySQL server
<code>--agentservicename</code>	Service name for the Agent
<code>--agentuser</code>	Username of the agent for connecting to the monitored MySQL server

Option Name	Description
<code>--checkmysqlhost</code>	Validate the supplied MySQL hostname
<code>--createBackup</code>	(Upgrade only) Create backup.
<code>--debuglevel</code>	Set the debug information level
<code>--debugtrace</code>	File for a debug trace of the installation
<code>--generalpassword</code>	General user password for the <code>--generaluser</code>
<code>--generaluser</code>	General user username for the monitored MySQL server
<code>--help</code>	Display the list of valid options
<code>--installdir</code>	Installation directory
<code>--installer-language</code>	Language selection
<code>--limitedpassword</code>	Limited user password for the <code>--limiteduser</code>
<code>--limiteduser</code>	Limited user username for the monitored MySQL server
<code>--managerhost</code>	Hostname of IP address of the MySQL Enterprise Monitor server
<code>--managerport</code>	TCP/IP port of the MySQL Enterprise Monitor server
<code>--mode</code>	Installation mode
<code>--mysql-identity-source</code>	MySQL instance identify definition
<code>--mysqlconnectiongroup</code>	Sets the group for the provided MySQL connection
<code>--mysqlconnmethod</code>	Connection method to the monitored MySQL server
<code>--mysqlhost</code>	MySQL hostname/IP address
<code>--mysqlpassword</code>	MySQL password for the monitored <code>--mysqluser</code> .
<code>--mysqlport</code>	TCP/IP port for the monitored MySQL server
<code>--mysqlsocket</code>	Unix socket/Named pipe for the monitored MySQL server
<code>--mysqluser</code>	MySQL Administrative username for the monitored MySQL server
<code>--optionfile</code>	Installation option file
<code>--restartImmediately</code>	(Upgrade only) Restart Agent immediately after updating all files.
<code>--unattendedmodeui</code>	Unattended mode user interface
<code>--version</code>	Display the product information

- `--agentpassword`

Command-Line Format	<code>--agentpassword</code>
Type	String

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

- `--createBackup`

Command-Line Format	<code>--createBackup</code>
Type	Boolean
Default Value	1

Whether to backup the data.



Note

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

- `--restartImmediately`

Command-Line Format	<code>--restartImmediately</code>
Type	Boolean
Default Value	1

Restart Agent immediately after updating all files.



Note

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

- `--agentuser`

Command-Line Format	<code>--agentuser</code>
Type	String
Default Value	<code>agent</code>

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

- `--checkmysqlhost`

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

Validate the MySQL hostname or IP address

- `--debuglevel`

Command-Line Format	<code>--debuglevel</code>
Type	Numeric
Default Value	2
Minimum Value	0

Maximum Value	4
---------------	---

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

- `--debugtrace`

Command-Line Format	<code>--debugtrace</code>
Type	String

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

- `--installdir`

Command-Line Format	<code>--installdir</code>
Type	String
Default Value (OS X)	<code>/Applications/mysql/enterprise/agent/</code>
Default Value (Unix)	<code>/opt/mysql/enterprise/agent/</code>
Default Value (Windows)	<code>C:\Program Files\MySQL\Enterprise Agent</code>

Specify the directory into which to install the software.

- `--installer-language`

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

Set the language to use for the installation process.

- `--managerhost`

Command-Line Format	<code>--managerhost</code>
Type	String

The hostname or IP address of MySQL Enterprise Service Manager.

- `--managerport`

Command-Line Format	<code>--managerport</code>
Type	Numeric
Default Value	<code>18443</code>

- `--mode`

Command-Line Format	<code>--mode</code>
Type	String
Default Value (Linux)	<code>gtk</code>
Default Value (OS X)	<code>osx</code>
Default Value (Unix)	<code>xwindow</code>
Default Value (Windows)	<code>win32</code>
Valid Values (Linux)	<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))
Valid Values (OS X)	<code>osx</code> (Mac OS X (native)) <code>text</code> (Text (command-line)) <code>unattended</code> (Unattended (no dialogs/prompts))
Valid Values (Unix)	<code>xwindow</code> (X Windows (native)) <code>text</code> (Text (command-line)) <code>unattended</code> (Unattended (no dialogs/prompts))
Valid Values (Windows)	<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>
Type	String
Default Value	<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>
Type	String
Default Value	127.0.0.1

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

- `--mysqlpassword`

Command-Line Format	<code>--mysqlpassword</code>
Type	String

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

- `--mysqlport`

Command-Line Format	<code>--mysqlport</code>
Type	Numeric
Default Value	3306

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

- `--mysqlsocket`

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

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

- `--mysqluser`

Command-Line Format	<code>--mysqluser</code>
Type	String

An MySQL Server administrative user for the MySQL instance to monitor. This user must already exist.

- `--agent_autocreate`

Command-Line Format	<code>--agent_autocreate</code>
Type	Boolean

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

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

- `--generaluser`

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

Type	String
------	--------

The username for the general user.

- `--generalpassword`

Command-Line Format	<code>--generalpassword</code>
Type	String

Password for the `--generaluser`.

- `--limiteduser`

Command-Line Format	<code>--limiteduser</code>
Type	String

The username for the limited user.

- `--limitedpassword`

Command-Line Format	<code>--limitedpassword</code>
Type	String

Password for the `--limitedpassword`.

- `--optionfile <optionfile>`

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

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

- `--unattendedmodeui`

Command-Line Format	<code>--unattendedmodeui</code>
Type	String
Default Value	<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>
Type	String
Default Value	<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.

- `--mysqlconnectiongroup`

Command-Line Format	<code>--mysqlconnectiongroup</code>
Type	String

Optionally sets the MySQL instance group for the connection.

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

- `--agentservicename`

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

When 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 10 Performance Tuning MySQL Enterprise Monitor

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.

10.1 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 10.1 Apache Tomcat Environment File Location (default)

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

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

Table 10.2 Installation Parameters

Parameter	Small	Medium	Large
Tomcat Heap Size	512MB	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 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 MySQL Enterprise Service Manager.

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

UNIX, Linux, and Mac

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

Microsoft Windows

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

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

Tuning InnoDB Memory

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 repository can be found in the following locations:

Table 10.3 MySQL Enterprise Service Manager repository configuration file location (default)

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

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

10.2 Tuning CPU

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

To do this, you should:

1. Stop the application Apache Tomcat server and MySQL Instance.

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

Update the configuration with the configuration tool. This tool can be found at the following location:

Table 10.4 Configuration utility location (default)

Operating System	Path
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\bin\config.bat</code>
Linux / Unix	<code>/opt/mysql/enterprise/monitor/bin/config.sh</code>
macOS	<code>/Applications/mysql/enterprise/monitor/bin/config.sh</code>

For more information on the configuration utilities, see [Chapter 11, Configuration Utilities](#).

10.3 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 10.5 Apache Tomcat configuration file location (default)

Operating System	Path
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\conf\server.xml</code>
Linux / Unix	<code>/opt/mysql/enterprise/monitor/apache-tomcat/conf/server.xml</code>
macOS	<code>/Applications/mysql/enterprise/monitor/apache-tomcat/conf/server.xml</code>

Edit the following section:

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

A good baseline to test is 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.

10.4 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 heap size must be increased by 64MB for every 10 additional MySQL instances.

The agent heap size is defined in the `MEM_AGENT_JAVA_OPTS` parameter in the `setenv.sh/bat` file.

**Note**

The default heap size is 128MB.

- The `data-reporting-threads` parameter must be increased by 2 for every 15-20 MySQL instances monitored.

The `data-reporting-threads` parameter is defined in the agent's `bootstrap.properties` file.

Chapter 11 Configuration Utilities

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

11.1 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 14.1, “Initial Log-In”](#).



Important

These parameters enable you to configure MySQL Enterprise Service Manager from script or command line and should only be run while the MySQL Enterprise Service Manager is stopped.

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

Table 11.1 Default Location of Agent Configuration Utility

Operating System	File Location
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\bin\config.bat</code>
UNIX/Linux	<code>/opt/mysql/enterprise/monitor/bin/config.sh</code>
macOS	<code>/Applications/mysql/enterprise/monitor/bin/config.sh</code>

Use `--help` to view the options.

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

- [Service Manager Configuration Utilities](#): define or change the configuration of MySQL Enterprise Service Manager.
- [Service Manager Certificate Utilities](#): modify or upgrade 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 11.2 Service Manager Configuration Utilities

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

Name	Description
<code>--mysql-port=<value></code>	MySQL port for the Service Manager repository. The port the target MySQL server listens on. The default is 13306.
<code>--mp=<value></code>	
<code>--mysql-db=<value></code>	MySQL database for the Service Manager repository. The name of the database used for the repository. The default is <code>mem</code> .
<code>--md=<value></code>	
<code>--mysql-server=<value></code>	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.
<code>--ms=<value></code>	
<code>--mysql-check-requirements</code>	Checks your repository to ensure compatibility with MySQL Enterprise Service Manager. For more information on the checks performed, see Section 4.2.4, "MySQL Enterprise Monitor Repository" .
<code>--mysql-upgrade</code>	Runs <code>mysql_upgrade</code> on the repository defined in <code>--mysql-basedir</code> .
<code>--mysql-basedir</code>	Base directory of the local MySQL installation.
<code>--sm-admin-user=<value></code>	Service Manager manager username. The user defined here is added to the manager role.
<code>--sm-agent-user=<value></code>	Service Manager agent username. The user defined here is added to the agent role.
<code>--auto-update</code>	Enable automatic checking for online updates.
<code>--purge-quant=<value></code>	Defines the Query Analyzer data retention policy. Query Analyzer data older than the number of days defined here is deleted. Default is 28 days.
<code>--purge-data=<value></code>	Defines the historical data retention policy. Historical data older than the number of days defined here is deleted. Default is 28 days.
<code>--proxy-host=<value></code>	HTTP Proxy host.
<code>--proxy-port=<value></code>	HTTP Proxy port.
<code>--proxy-user=<value></code>	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 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

The certificate utilities enable you to modify, renew, import, and upgrade your SSL certificates.



Important

It is strongly recommended you use this utility to modify your SSL certificates and not attempt to modify them with any other tool.

Table 11.3 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 non-OS X platforms. That is, Microsoft Windows, Linux, and so on.
<code>--upg</code>	
<code>--upgrade-osx</code>	Upgrades certificates on OS X platforms.
<code>--upgo</code>	
<code>--new-install</code>	Generates a keystore with a new self-signed certificate for a new installation
<code>--ni</code>	
<code>--accept-keystore-password</code>	If specified, the user is prompted to enter the keystore password. If not specified, the default password <code>changeit</code> is used to access the keystore.
<code>--akp</code>	
<code>--renew</code>	Renew an existing, self-signed certificate. If the certificate is not self-signed, an error is returned.
<code>--import-certificate=<value></code>	Imports the specified certificate. For example: <code>--import-certificate=/path/to/client.crt</code>
<code>--import-key=<value></code>	Imports the specified private key. For example: <code>--import-key=/path/to/client.key</code>

11.2 Agent Configuration Utility

The `agent.sh/agent.bat` script is used to configure an Agent. The following table lists the default locations for each supported operating system:

**Important**

These parameters enable you to configure MySQL Enterprise Monitor Agent from script or command line and should only be run while the MySQL Enterprise Monitor Agent is stopped.

Table 11.4 Default Location of Agent Configuration Utility

Operating System	File Location
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Agent\bin\agent.bat</code>
UNIX/Linux	<code>/opt/mysql/enterprise/agent/bin/agent.sh</code>
macOS	<code>/Applications/mysql/enterprise/agent/bin/agent.sh</code>

Use `--help` to view its options.

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

- **Agent Connection Utilities:** test or edit the connections of MySQL Enterprise Monitor Agent to the monitored MySQL instance.
- **Agent Configuration Utilities:** configure or edit the connection from MySQL Enterprise Monitor Agent to MySQL Enterprise Service Manager.

**Important**

It is not possible to run `agent.sh` from the command line as `root`, but only as `mysql`.

Agent Connection Utilities

The following parameters enable you to create, test, and edit the connections from MySQL Enterprise Monitor Agent to the monitored MySQL instance.

Table 11.5 Agent Connection Utility

Name	Description
<code>--test-credentials</code>	Test the MySQL connection credentials.
<code>-T</code>	
<code>--test-privileges</code>	Test the admin user's privileges to manage other users.
<code>--create-connection</code>	Create or modify a MySQL connection.
<code>-c</code>	
<code>--delete-connection</code>	Closes and deletes a MySQL connection. This parameter requires <code>--connection-id</code> .
<code>-d</code>	
<code>--show</code>	Show information about all MySQL connections on this agent
<code>-s</code>	
<code>--auto-manage-extra-users</code>	Auto-create general / limited users (Actions: Create, Modify)
<code>-m</code>	
<code>--host=<value></code>	Host for the MySQL instance (Actions: Create, Modify)

Name	Description
<code>-h <value></code>	
<code>--port=<value></code>	Port for the MySQL instance (Actions: Create, Modify)
<code>-P <value></code>	
<code>--socket=<value></code>	Socket for the MySQL instance (Actions: Create, Modify)
<code>-S <value></code>	
<code>--limited-user=<value></code>	Limited level credentials (Actions: Create, Modify)
<code>-l <value></code>	
<code>--general-user=<value></code>	General user credentials
<code>-k <value></code>	
<code>--admin-user=<value></code>	Admin user credentials
<code>-j <value></code>	
<code>--connection-id=<value></code>	Connection ID
<code>-i <value></code>	
<code>--connection-group=<value></code>	Specify the name of the group to add this connection to. If the named group does not exist, it is created and the connection added to it. This parameter can be used multiple times in the <code>--create-connection</code> command. The connection is added to each group named.
<code>-g <value></code>	
<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.
<code>--mysql-identity-</code> <code>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

The following parameters enable you to configure or edit the connection from MySQL Enterprise Monitor Agent to MySQL Enterprise Service Manager.

Table 11.6 MySQL Enterprise Monitor Agent Configuration Utility

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

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

Chapter 12 Uninstalling MySQL Enterprise Monitor

Removing MySQL Enterprise Monitor requires removing MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agents.

12.1 Windows Platforms

Removing MySQL Enterprise Service Manager

On Microsoft Windows 7 and 8, remove MySQL Enterprise Service Manager by opening the **Control Panel** and choosing **Uninstall a Program**. Locate the entry for **MySQL Enterprise Monitor** and select **Uninstall** from the Control Panel toolbar.

On Microsoft Windows 10, remove MySQL Enterprise Service Manager by opening **Settings** and selecting **Apps & Features**. Locate the entry for **MySQL Enterprise Monitor** and select **Uninstall**.

The uninstall process prompts you to save existing data and log files. Choose this option if you plan to reinstall at a later time.

If you do not want to backup the data, you can delete the `C:\Program Files\MySQL\Enterprise\Monitor` directory after removing MySQL Enterprise Service Manager.

If you want to retain the backup, do not remove the `C:\Program Files\MySQL\Enterprise\Monitor` directory. Doing so will delete these files.

Removing MySQL Enterprise Monitor Agent

On Microsoft Windows 7 and 8, remove MySQL Enterprise Monitor Agent by opening the **Control Panel** and choosing **Uninstall a Program**. Locate the entry for **MySQL Enterprise Monitor Agent** and select **Uninstall** from the Control Panel toolbar.

On Microsoft Windows 10, remove MySQL Enterprise Monitor Agent by opening **Settings** and selecting **Apps & Features**. Locate the entry for **MySQL Enterprise Monitor Agent** and select **Uninstall**.



Warning

To remove only one of the agents from a machine hosting several agents, do not remove the `MySQL Enterprise Monitor Agent` entry from the **Add or Remove Programs** menu. To remove a single agent, see [Removing a Single Agent](#).

Removing MySQL Enterprise Monitor Agent automatically deletes its associated `.log` and `.pid` files. After removing the Monitor Agent, you might need to remove the directories, `C:\Program Files\MySQL\Enterprise` and `C:\Program Files\MySQL\Enterprise\Agent`.

Removing MySQL Enterprise Monitor Agent in this way also removes the agent service. If you are running additional agents on the same server, you must remove them agents manually. See [Removing a Single MySQL Enterprise Monitor Agent](#) for instructions on uninstalling individual agents.

Removing a Single MySQL Enterprise Monitor Agent

To remove only one of the agents from a machine that is running several agents, do not remove the `MySQL Enterprise Monitor Agent` entry from the **Add or Remove Programs** menu. To remove a single agent, do the following:

1. Stop the agent.

You must stop the agent before attempting to remove it; for instructions on stopping an agent see, [Section 6.5.1, “Starting/Stopping the Agent on Windows”](#).

2. Confirm the location of the log files.

To confirm the location of the agent log files, check the `ini` file.

3. Remove the agent as a service. This can only be done from the command line.

Remove MySQL Enterprise Monitor Agent as a Windows service with the following command:

```
$> sc delete AgentName
```



Note

To find the AgentName, from the **Services** pane of the Microsoft Management Console, open the **Properties** page for the MySQL Enterprise Monitor Agent service. The **Service name** field on the **General** tab contains the required value.

To confirm the agent has been removed, check the **Services** pane of the Microsoft Management Console.

4. Remove or archive any log or configuration files associated with the agent.

12.2 macOS Platforms

Removing MySQL Enterprise Service Manager

To uninstall MySQL Enterprise Service Manager, run the `uninstall.app` located in the root directory of your MySQL Enterprise Service Manager installation.

The uninstall process prompts you to save existing data and log files. Choose this option if you plan to reinstall at a later time.

If you do not want to backup the data, you can delete the installation directory after uninstalling MySQL Enterprise Service Manager.

Removing MySQL Enterprise Monitor Agent

To remove MySQL Enterprise Monitor Agent, navigate to the `init.d` directory and issue the following command:

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

Run the `uninstall.app` file located in the agent's installation directory.

Removing the Monitor Agent automatically deletes its associated `.log` and `.pid` files. After uninstalling MySQL Enterprise Monitor Agent, you can remove the installation directory.

Removing a Single MySQL Enterprise Monitor Agent

To uninstall only one of the agents from a machine that is running several agents, do not run the uninstall program. To uninstall a single agent and leave other agents in place, do the following:

1. Stop the agent.

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

It is best to stop the agent before removing it; for instructions on stopping an agent, see [Section 6.5.2, “Starting/Stopping the Agent on macOS”](#).

To confirm the location of the agent log files, check the `.ini` file.

You can then remove the agent as a daemon by removing its entry in the `init.d` directory.

Also remove or archive any log or configuration files associated with this agent.

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

12.3 Unattended Uninstallations

This section describes how to uninstall MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent as an unattended process. The unattended uninstallation can be run from the command line.

Both MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent have identical uninstallation options. To display those options, from the command line run the `uninstall` file in your installation directory, with the `--help` option.

The following options are available:

Table 12.1 MySQL Enterprise Monitor Uninstaller Options

Option	Description
<code>--help</code>	Displays the list of options.
<code>--version</code>	Displays the product name and version.
<code>--debuglevel <debuglevel></code>	Sets the verbosity of the uninstallation log. 0 is the lowest, 4 the highest, and 2 is the default.
<code>--mode <mode></code>	<p>Sets the uninstallation mode. This varies according to the platform. For example, on Linux-based systems, you can choose a GUI-based uninstaller with <code>--mode gtk</code>, or choose a text-only, console-based uninstallation with <code>--mode text</code>.</p> <p>The following is a list of the GUI-based uninstallation options available:</p> <ul style="list-style-type: none"> • Windows: <code>Win32</code> • OS X: <code>osx</code> • Solaris: <code>xwindow</code> • Linux: <code>gtk</code> (Default) and <code>xwindow</code>. <p><code>--mode</code> can also initiate text mode and unattended uninstallations.</p> <ul style="list-style-type: none"> • <code>--mode text</code>: starts a text-only, console-based uninstallation process. Text-based uninstallation is not available on Windows platforms. • <code>--mode unattended</code>: starts an unattended uninstallation.

Option	Description
<code>--debugtrace</code> <code><debugtrace></code>	Sets the path and filename of the uninstallation log file.
<code>--installer-language</code>	Sets the language of the uninstallation. Possible values are: <ul style="list-style-type: none">• <code>en</code>: English. Default value.• <code>ja</code>: Japanese.

Unattended Uninstallation

To run an unattended uninstallation process, in which no dialogs, prompts or warnings are displayed, run the following command in the installation directory of your MySQL Enterprise Service Manager or MySQL Enterprise Monitor Agent:

```
$>./uninstall --mode unattended
```

Part III Configuring MySQL Enterprise Service Manager

Table of Contents

13	Configuring MySQL Enterprise Service Manager	101
14	User Interface	103
14.1	Initial Log-In	103
14.2	Setting the Timezone and Locale	104
14.3	Menus and Toolbars	104
14.3.1	Target Selection	104
14.3.2	Navigation Menus	105
14.3.3	User Menu	107
14.3.4	Status Summary	108
14.3.5	System and User-defined Filters	108
14.3.6	Configuration Views	109
15	Global Settings	111
15.1	Server Locale	111
15.2	Server Hostname	111
15.3	Customize MySQL Server Name	111
15.4	Data Purge Behavior	113
15.5	HTTP Proxy Settings	114
15.6	External Authentication	114
16	Manage Instances	119
16.1	Manage Instances Controls	119
16.2	Monitored Instance Details	121
16.3	Adding Instances	123
16.3.1	Adding a MySQL Instance	123
16.3.2	Adding Multiple MySQL Instances	127
16.4	Monitoring Amazon RDS	127
16.5	Filtering MySQL Instances	128
17	Asset Management	131
17.1	Agents View	131
17.2	Hosts View	132
17.3	NDB Clusters View	134
17.4	Deleting Assets	135
17.5	Deleting Assets from the Command Line	138
18	Managing Groups of Instances	145
19	Advisors	149
19.1	Manage Advisors Page	149
19.2	Advisor Types	153
19.3	Advisor Thresholds	154
19.4	Advisor Schedules	156
20	Expression-Based Advisor Reference	157
20.1	Administration Advisors	157
20.2	Agent Advisors	163
20.3	Availability Advisors	163
20.4	NDB Cluster Advisors	165
20.5	Memory Usage Advisors	166
20.6	Monitoring and Support Services Advisors	168
20.7	Operating System Advisors	169
20.8	Performance Advisors	169
20.9	Replication Advisors	175
20.10	Schema Advisors	180
20.11	Security Advisors	184
21	GUI-Based Advisor Reference	191

21.1 Agent Health Advisor	191
21.2 MySQL Enterprise Backup Health Advisor	194
21.3 MySQL Process Discovery Advisor	194
21.4 Duplicate Host Identity	196
21.5 Duplicate MySQL Server UUID	196
21.6 sys Schema Install Advisor	196
21.7 CPU Utilization Advisor	196
21.8 Filesystem Free Space Advisor	198
21.9 MySQL Process	200
21.10 Query Analysis Advisors	200
21.11 Group Replication Advisors	202
21.12 Security Advisors	204
21.13 NDB Cluster GUI Advisors	206
22 Event Handlers	211
22.1 Event Handlers	211
22.1.1 Event Handlers	211
22.1.2 Event Handlers Page	211
22.2 Creating Event Handlers	217
22.2.1 Event Action Log	220
22.2.2 Suspending an Event Handler	220
23 Access Control	221
23.1 Users and Roles	221
23.2 Permissions	221
23.3 Monitored Assets Permissions	222
23.3.1 Server Group & MySQL NDB Cluster	223
23.3.2 MySQL Instances	223
23.3.3 MySQL NDB Cluster Permissions	225
23.4 Monitoring Services	225
23.5 MySQL Enterprise Monitor	226
23.6 Default Users and Roles	228
23.7 Creating Users and Roles	229
24 Access Control - Best Practices	231
24.1 Open Permission Sets	232
24.2 Strict Permission Set	233

Chapter 13 Configuring MySQL Enterprise Service Manager

This part describes the configuration of MySQL Enterprise Service Manager. The following topics are described:

- [Chapter 14, *User Interface*](#): first time setup and an overview of the user interface.
- [Chapter 15, *Global Settings*](#): how to define locales, hostnames, external authentication, and so on.
- [Chapter 16, *Manage Instances*](#): how to add instances, edit connections, delete instances and manage bad connections, unmonitored instances, or unreachable agents.
- [Chapter 18, *Managing Groups of Instances*](#): how to create and manage groups of assets.
- [Chapter 19, *Advisors*](#): description of advisors and their purpose.
- [Chapter 20, *Expression-Based Advisor Reference*](#) and [Chapter 21, *GUI-Based Advisor Reference*](#): descriptions of the default advisors and their purpose
- [Chapter 23, *Access Control*](#): users, roles, and permissions.

Chapter 14 User Interface

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

14.1 Initial Log-In

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

Figure 14.1 Welcome to MySQL Enterprise Monitor page

Welcome to MySQL Enterprise Monitor

To complete installation and configuration, please complete the form below.

Create user with 'manager' role

There must be at least one user with the 'manager' role at all times, with full access to all privileges and assets. Keep these credentials safe, further users and roles can be created once complete.

Username

Password

Confirm Password

Create user with 'agent' role

Agents require permission to access the Service Manager, this user will be created with the 'agent' role, and can be used when configuring new Agent connections to the Service Manager.

Username

Password

Confirm Password

Configure 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

Configure data retention settings

How long collected data should be stored before being overwritten or removed.

Remove Non-Aggregated Metric Data Older Than

Remove Hourly Aggregated Metric Data Older Than

Remove Daily Aggregated Metric Data Older Than

Remove Query Analyzer Data Older Than

You must perform the following tasks:

- **Create user with Manager role:** the manager user is used for the initial session, configuring MySQL Enterprise Service Manager, and defining the access control list. The user defined here is added to the default `manager` role. For more information on the default roles, see [Chapter 23, Access Control](#).

This creates the Manager user described in [Section 4.3.3, “Users Created on First Log-in”](#). Record the user name and password, as these credentials are required for any future login.

- **Create user with agent role:** the Agent user credentials are used by every monitoring agent to connect to MySQL Enterprise Service Manager. The user defined here is added to the default `agent` role. For more information on the default roles, see [Chapter 23, Access Control](#).

This is the user described in [Section 4.3.3, “Users Created on First Log-in”](#). The agent must connect to MySQL Enterprise Service Manager to transmit the monitored data. Record the agent's credentials; this information is required when installing the agent.



Note

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

- **Configure Data Retention Settings:** define how long collected data must be retained. For more information on data purging, see [Section 15.4, “Data Purge Behavior”](#).

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 Online Updates:** enables checking for updates online. If your organization uses a HTTP proxy, you must check the Use HTTP Proxy field, and complete the fields displayed when this is enabled.

Click the **Complete Setup** button.



Important

These settings can also be defined from the command line, or in a script. For more information, see [Chapter 11, Configuration Utilities](#).

14.2 Setting the Timezone and Locale

If this is the first time launching the MySQL Enterprise Monitor User Interface, you must 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

Set the time zone (and the system clock) correctly because this setting affects how MySQL Enterprise Service Manager's graphs display data.

The locale determines the default language displayed when logging in to the MySQL Enterprise Monitor User Interface. This selection overrides the default browser settings whenever this specific user logs in.

At this point MySQL Enterprise Service Manager's repository is being monitored, and the built-in agent is attempting to auto-discover additional MySQL instances on the host.

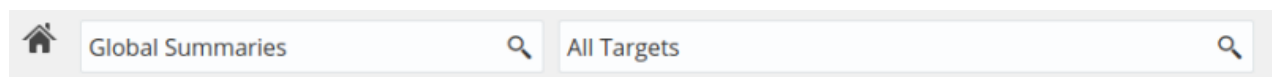
14.3 Menus and Toolbars

This section describes the menus and toolbars of MySQL Enterprise Monitor's user interface.

14.3.1 Target Selection

This section describes the target selection menus.

Figure 14.2 Target Selectors



The target selection menus enable you to filter the information displayed in the selected view. You can filter on individual assets, such as instances or hosts, or you can filter by groups and topologies, viewing information on all, or one, of the contents of the selected group, topology, or cluster. The filter uses the following criteria:

- **Global Summaries:** select one of the following possible groupings:
 - **Groups:** filter by group. Select a group to view only the information collected for the selected group's assets.
 - **Replication Topologies:** filter by replication topology. Select a topology to view only the information collected for the selected topology's assets.
 - **InnoDB Clusters:** filter by InnoDB cluster.



Note

Group Replications are also listed under **InnoDB Clusters**.

- **NDB Clusters:** filter by NDB cluster.
- **All Targets:** select one of the assets contained by the selection made in Global Summaries, or any one of the monitored assets if no selection was made in Global Summaries.



Note

You can select the required asset or group from the list, or search for the asset by typing the first few characters of the asset's name. The list is filtered based on the characters entered.

To cancel the changes and reset the filters, press Esc.



Note

If the asset selection is not supported by the view, a message is displayed explaining why it cannot be displayed. For example, if you select a group, while viewing a Topology, the following message is displayed: The current selection does not support the Topology view. Please select a Replication or Cluster topology.

14.3.2 Navigation Menus

This section describes the navigation menus of the user interface.



Note

The navigation menus are context sensitive. That is, they are displayed according to the selection made in the **View Filters**. For example, the **Metrics** menu, **User Statistics**, is displayed only if a MySQL instance is selected, not if a host, agent, or **All Targets** is selected

Overview

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 25, Overview](#). This is the first page displayed after the initial setup steps are completed.

Events

The **Events** page lists all the events for the monitored assets to which you have access. See [Figure 26.1, “Events Page with Filter”](#) for more information.

Metrics

The **Metrics** menu contains the following links:

- **All Timeseries**: opens the **All Timeseries** graphs page. See [Section 30.1, “All Timeseries Graphs”](#) for more information.
- **Table Statistics**: opens the **Table Statistics** report. This report enables you to view detailed table statistics in both table and treemap forms. This page utilizes the `sys` schema and is only supported on MySQL versions 5.6 and 5.7.

See [Section 30.4, “Table Statistics”](#) for more information.

- **User Statistics**: opens the **User Statistics** report. This report enables you to view detailed information on the database users. This page utilizes the `sys` schema and is only supported on MySQL versions 5.6 and 5.7.

See [Section 30.5, “User Statistics”](#) 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 utilizes 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 30.7, “InnoDB Buffer Pool Usage”](#) for more information.

- **Processes**: opens the **Processes** report page. The Processes report lists the processes currently running on the selected MySQL instance. See [Section 30.3, “Processes Report”](#) for more information.

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

- **Lock Waits**: opens the **Lock Waits** report page. See [Section 30.2.3, “Lock Waits Report”](#) for more information.

This page utilizes 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 30.7, “InnoDB Buffer Pool Usage”](#) for more information.

Statements

Opens the Query Analyzer page. See [Chapter 32, Query Analyzer View](#) for more information.

Replication

Replication: opens the **Replication** view. This provides a detailed view of the current state of your monitored replication servers. For more information, see [Chapter 28, Replication Dashboard](#).

Backups

Backups: opens the **Backup** view. This provides a detailed view of the configured backups of your monitored instances. For more information, see [Chapter 27, Enterprise Backup Dashboard](#).

Configuration

- **Instances:** opens the **Instances** view. This 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 16, Manage Instances](#).
- **Groups:** opens the **Manage Groups** page. For more information, see [Chapter 18, Managing Groups of Instances](#)
- **Advisors:** opens the **Advisors** page. See [Chapter 19, Advisors](#) for more information.
- **Event Handlers:** opens the **Event Handlers** page. See [Chapter 22, Event Handlers](#) for more information.
- **Users:** opens the **Users** page. See [Chapter 23, Access Control](#) for more information.
- **Roles:** opens the **Roles** page. See [Chapter 23, Access Control](#) for more information.
- **Settings:** opens the **Settings** page. See [Chapter 15, Global Settings](#) for more information.

Help

- **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”](#).

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 canceled and the defined refresh behavior resumes.

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

14.3.3 User Menu

This section describes the contents of the User menu.



Note

This menu is renamed according to the user logged in. If the user is named `admin`, the user menu is labeled **admin**.

- **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.

14.3.4 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 14.3 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.

14.3.5 System and User-defined Filters

To create a user filter, you can either create a filter and save it by clicking **Save as...** menu item, or create a new filter by clicking **New**, defining your filter criteria, and clicking **Save as....**

You can also create a filter by using an existing filter as a template. Select the filter and make your changes. If you create a named filter based on an existing filter, the **-clone** is appended to the name when you edit the new filter. The name can be edited as required.

System filters are listed with a padlock on the left side of their name. These cannot be edited, but can be used as templates for new filters.

To set a filter as the default, select it in the drop-down list, and select **Set as Default** from the adjacent drop-down menu. A star is displayed next to the default filter.



Note

If you upgraded from a previous version, and used default filters in that version, your filters are migrated and renamed `User Default`.

Named filters are available on the following pages:

- **MySQL Instances:** contains the following system filters:
 - **All MySQL Instances:** this is the default selection. All instances are displayed.
 - **MySQL Instances Monitored By Old Agents:** only displays MySQL instances monitored by agents older than the current version.
 - **MySQL Instances Using EOL Versions:** displays only the End Of Life MySQL instances monitored by this installation.
- **Events:** contains the following system filters:
 - **All Open Events:** (default) displays all open events.
 - **All Availability Events:** filters on the **Availability (any)** advisor selection. Only events generated by the availability advisors are displayed.
 - **Open Emergency Events:** filters on the current status of **Emergency**. Only Emergency events are displayed.
 - **Open Events with Status of Warning or Greater:** filters on the current status of **>=Warning**. Only events with status of Warning or higher are displayed.
- **All Timeseries Graphs:** the default filter is **All Timeseries Graphs**. No filtering, all graphs are displayed.
- **Advisors:** the default filter is **All Advisors**. No filtering, all advisors are displayed.

14.3.6 Configuration Views

Configuration views are similar to System and User Filters, see [Section 14.3.5, “System and User-defined Filters”](#), but are configured using a configuration form filter.

Configuration View is available on the following:

- **Query Analyzer** view. This view contains the following system configuration views:
 - **Administration Statements:** filters on statements of type **GRANT, REVOKE, RESET, SET, SHOW, FLUSH, CACHE, KILL, and SHUTDOWN**.
 - **All Statements:** (default) no filtering defined. All statements are displayed.
 - **DDL Statements:** filters on statements of type **CREATE, DROP, ALTER, TRUNCATE, and RENAME**.
 - **DML Statements:** filters on statements of type **SELECT, INSERT, UPDATE, DELETE, REPLACE, CALL, LOAD, DO, and HANDLER**.
 - **Prepared Statements:** filters on statements of type **PREPARE, EXECUTE, and DEALLOCATE**.

- **Replication Statements:** filters on statements of type **START**, **STOP**, **RESET**, and **CHANGE**
- **Statements with Errors:** filters on the advanced filter options of **Total Errors > 0**.
- **Statements with Full Table Scans:** filters on the advanced filter options of **Table Scan** notices and **Total Table Scans > 0**.
- **Statements with Max Exec Time Over 1 Second:** filters on the advanced filter options of **Max Exec Time > 1**.
- **Statements with Temporary Tables:** filters on the advanced filter options of **Total Temporary Tables > 0**.
- **Statements with Temporary Tables on Disk:** filters on the advanced filter options of **Total Temporary Disk Tables > 0**
- **Statements with Warnings:** filters on the advanced filter options of **Total Warnings > 0**
- **Table Maintenance Statements:** filters on statements of type **OPTIMIZE**, **ANALYZE**, **CHECK**, **REPAIR**, and **CHECKSUM**.
- **Transactional and Locking Statements:** filters on statements of type **BEGIN**, **COMMIT**, **ROLLBACK**, **SAVEPOINT**, **RELEASE**, **LOCK**, and **UNLOCK**.
- **Agents** view. This view contains the following system configuration views:
 - **All Agents:** all agents, regardless of status.
 - **Down Agents:** lists only those agents which are stopped or unreachable.
 - **Reachable Agents:** lists only those agents which are reachable.
- **Hosts** view. This view contains the following system configuration views:
 - **All Hosts:** lists all hosts currently monitored by MySQL Enterprise Service Manager.
- **NDB Clusters** view. This view contains the following system configuration views:
 - **All Clusters:** lists all NDB Clusters currently monitored by MySQL Enterprise Service Manager.

To create a configuration view, click **Configuration View**. The **Configuration View** form is displayed. Add properties as required using the **Add** button. Clear all properties using the **Reset** button. Or undo your changes by clicking **Discard Changes**.

You can create a configuration view, based on an existing view, by selecting the existing view from the **Select Configuration View** drop-down list and editing as required. As you edit the existing view, **-clone*** is appended to the view's name, indicating it is a temporary view. To save this customized view, select **Save as...** from the **Actions** menu.

You can also create a new, blank configuration view by selecting **New** from the **Actions** menu.

To save a new configuration view, select **Save as...** from the **Actions** menu. To save edits to an existing, custom, configuration view, select **Save** from the **Actions** menu.

To make a configuration view the default, select the required view, and select **Set as Default** from the **Actions** menu.

Chapter 15 Global Settings

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

15.1 Server Locale

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

15.2 Server Hostname

This section describes how to define your server hostname.

Figure 15.1 Server Hostname

Server Hostname

Hostname
my.mem.com
This hostname overrides the operating system hostname for use in notifications. It must be a resolvable hostname.

Port Number
18443
This port overrides the Apache Tomcat port for use in notifications.

Login Display Name
MEMServer1
This text overrides the hostname used on the login page. It's used as an alias for the operating system hostname.

Save

Table 15.1 Server Hostname Controls

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.

15.3 Customize MySQL Server Name

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

Figure 15.2 Customize MySQL Server Name

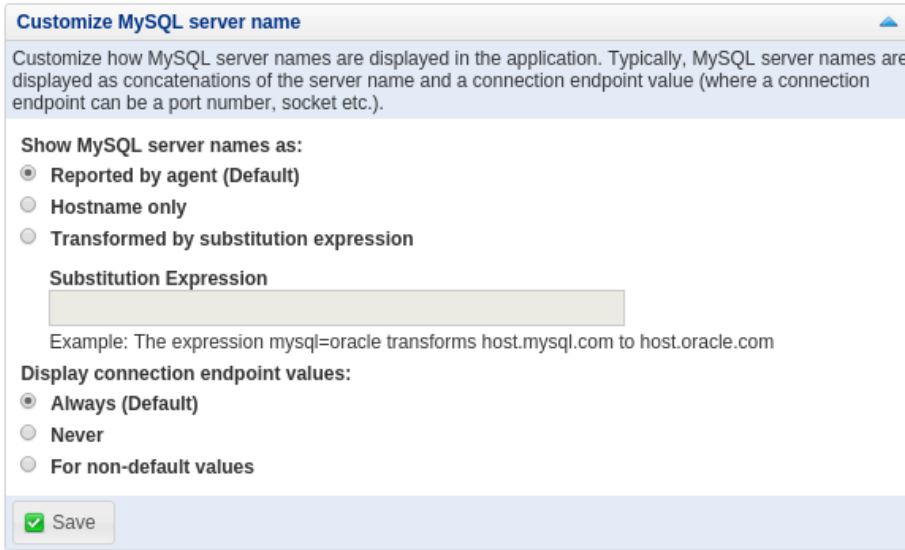


Table 15.2 Customize

Name	Description
Show MySQL Server Names as:	<p>Defines how hostnames are displayed. The following display settings are possible:</p> <ul style="list-style-type: none"> • 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 <code>companyname.com</code>. • 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	<p>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.</p> <p>You can include multiple, comma-separated, substitutions.</p> <p>For example:</p> <pre data-bbox="643 1598 1471 1780">dx521\.\example\.\com=Staging dx984[.\]example[.\]com=Production database-server-(.*?)\.\example\.\com=\$1 ^database-server-=" ", [.\]example[.\]com\$=" ",dx521="Staging DB"</pre> <p>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.</p>

Name	Description
Display Connection Endpoint Values	<p>Defines how connection endpoint values are displayed. The following display settings are possible:</p> <ul style="list-style-type: none"> • Always (default): endpoint values are always displayed. • Never: endpoint values are never displayed. • For non-default values: endpoint values are displayed only if they differ from <code>3306</code> and <code>/tmp/mysql.sock</code>.

15.4 Data Purge Behavior

Data Purge Behavior enables you to automatically remove old 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 15.3 Data Purge Behavior

The screenshot shows a configuration window titled "Data Purge Behavior". It contains four rows of settings, each with a text input field and a dropdown menu:

- Remove Non-Aggregated Metric Data Older Than:** Input field contains "4", dropdown menu is "Weeks".
- Remove Hourly Aggregated Metric Data Older Than:** Input field contains "1", dropdown menu is "Years".
- Remove Daily Aggregated Metric Data Older Than:** Input field contains "10", dropdown menu is "Years".
- Remove Query Analyzer Data Older Than:** Input field contains "4", dropdown menu is "Weeks".

At the bottom left of the window is a "Save" button with a blue arrow icon.

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 is 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 Non-Aggregated Metric Data Older Than:** configures the duration that the non-aggregated data collected from your monitored assets is retained. This includes all data collections, including CPU, memory and connections and activity statistics.
- **Remove Hourly Aggregated Metric Data Older Than:** configures the duration that the hourly aggregated data collected from your monitored assets is retained. This includes all data collections, including CPU, memory and connections and activity statistics.

- **Remove Daily Aggregated Metric Data Older Than:** configures the duration that the daily aggregated data collected from your monitored assets 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.

15.5 HTTP Proxy Settings

These settings were removed in MySQL Enterprise Monitor 8.0.32, the version that removed the integrated My Oracle Support (MOS) functionality.


Figure 15.4 HTTP Proxy Settings

The screenshot shows a dialog box titled "HTTP Proxy Settings". It has a title bar with a blue maximize button. Inside, there are four text input fields stacked vertically, labeled "Proxy Host:port", "Proxy Username", "Proxy Password", and "Confirm Password". At the bottom left, there is a "Save" button with a green checkmark icon.

15.6 External Authentication

Table 15.3 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 .



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 15.5 External Authentication Settings: LDAP

Table 15.4 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 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.

Name	Description
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 <code>sAMAccountName</code> , 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	<p>The authentication mode to use.</p> <ul style="list-style-type: none"> • 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	<p>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.</p> <p>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.</p>
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

Name	Description
	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 15.5 Active Directory Authentication

Name	Description
Domain	The Active Directory Domain.
Primary Server Hostname	Hostname of the Active Directory server to use.
Secondary Server Hostname	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 16 Manage Instances

The **Manage Instances** view presents information on the current connection status of all monitored instances. It also presents information on unsuccessful connections, unreachable agents, and MySQL instances which are not yet monitored.

To open the **Manage Instances** view, select **Instances** from the **Configuration** section of the navigation menu.

The **Instances** view 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 16.2, “Monitored 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](#).

16.1 Manage Instances Controls

This section describes the controls on the **Manage Instances** view.

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 view.



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 16.1 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 16.2 Unreachable Agents

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 check boxes and click **Monitor Instances**. The add instance dialog is displayed and is auto-populated with the agent name, instance address, and so on. For more information on adding connections, see [Section 16.3.1, “Adding a MySQL Instance”](#).

To ignore instances, make your selection and click Ignore Instances. A check box, **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** check box, 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 16.3 Unmonitored MySQL Instances

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.

16.2 Monitored Instance Details

The monitored instance section (untitled) 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 one of the following methods:

- Click the MySQL instance's *overflow menu* (three vertical dots), and choose **Edit Instance**
- Or select the check box on the left of the MySQL instance name, and click **Edit Instances**

The Edit Instances window is identical to the Add Instance Connection window described in [Section 16.3.1, "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 overflow menu.

Columns

The following columns are available:

Table 16.4 MySQL Instance Details

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 check box 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.
More	An overflow menu (three vertical dots) that offers context-specific options, such as Edit Instance.

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 MySQL Enterprise Service Manager.



Important

It is not possible to delete the 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 download the 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.

16.3 Adding Instances

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

- [Section 16.3.1, “Adding a MySQL Instance”](#)
- [Section 16.3.2, “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.

16.3.1 Adding a MySQL Instance

This section describes how to monitor a MySQL instance.

Connection Settings

The Connection Settings tab

Figure 16.1 Add Instance Connection Settings

Connection Settings | Encryption Settings | Advanced Settings | Group Settings

Monitor From [?]

Connect Using [?]

Instance Address [?] Port [?]


Admin User [?] Admin Password [?]


Auto-Create Less Privileged Users [?]

General User [?] General Password [?]

Limited User [?] Limited Password [?]

Table 16.5 Connection Settings Tab

Name	Description
Monitor From	<p>Select an Agent from the list of Agents to monitor this MySQL database Instance.</p> <p>It is recommend installing one Agent per Host and using that Agent to monitor all its MySQL Instances.</p> <p>Set up the Agent as a local connection by specifying TCP/IP and an Instance Address of 127.0.0.1, or use a socket file. If there is no local Agent on the Host and you are unable to install one, use the built-in or another Agent to monitor remotely.</p> <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 20px;">  </div> <div> <p>Note</p> <p>If the instance is monitored remotely, it is not possible to retrieve any information on the host. To monitor a host, an agent must be installed on that host.</p> </div> </div>
Connect Using	<p>Select TCP/IP or a socket to connect to the instance. Socket connections can only be used for an Agent that is installed on the same machine as the target instance, and do not work with instances running on Windows.</p>
Instance Address and Port	<p>The IP address, or valid hostname, and port number the instance is listening on. If the host/agent chosen is local to this instance, you should use 127.0.0.1 here.</p>

Name	Description
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	<p>Choose Yes to create the General and Limited users on the MySQL instance. You must add a user name and password for both. For more information on these users, see Section 6.2, "Creating MySQL User Accounts for the Monitor Agent".</p> <p>Choose No if you intend to use the Admin user for all data collection.</p> <div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;">  </div> <div> <p>Note</p> <p>It is strongly recommend to use the General and Limited user.</p> </div> </div>
General User and Password	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 16.2 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 16.6 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.

Name	Description
	 <p>Note</p> <p>This is not required if you are using a self-signed certificate.</p>

Advanced Settings

Figure 16.3 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 16.7 Advanced Settings

Name	Description
Discover Replication Topologies	<p>Specifies whether the agent attempts to discover if the instance is part of a replication group and discover the other members of that replication group.</p> <p>When performing replication topology discovery, the agent attempts to read the slave's <code>master.info</code>, and use the stored credentials to log in to the master to read its inventory table and retrieve the master's UUID.</p> <p>If set to No, no replication discovery is attempted.</p>
MySQL Identity Source	<p>Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.</p> <ul style="list-style-type: none"> • Default: uses either the <code>server_uuid</code> variable if present, or generates a random new UUID • Host Plus Datadir generates a hash of the host identity and the path to the MySQL instances data directory to create a unique identity. <p>  <p>Note</p> <p>Host Plus Datadir can be used only if the agent is running on the same host as the MySQL instance for this connection.</p> </p>

Name	Description
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**.

16.3.2 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.

16.4 Monitoring Amazon RDS

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



Important

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

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

When configuring a MySQL instance to monitor from the Instances view, 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 Instances view.



Note

MySQL Performance Schema is not enabled by default on Amazon RDS. If you intend to use Query Analyzer, you must enable Performance Schema by setting the `performance_schema` parameter to 1 in instance parameter group on the AWS console and restart the instance.

16.5 Filtering MySQL Instances

To search for specific instances, click the filter icon. The filter is displayed. For information on the named filters, see [Section 14.3.5, "System and User-defined Filters"](#).

Figure 16.4 MySQL Instance Filter

Table 16.8 MySQL Instance Filter

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

The **Agent Version** and **MySQL Version** fields support the use of range operators (>, <=), enabling you to define ranges of versions to filter on. For example, setting **MySQL Version** to <=5.1 returns all MySQL instances older than MySQL 5.1.



Note

Filtering on MySQL or Agent version uses a regular expression which does not support the use of partial version numbers, such as "5.". 5 or 5.6 return a result, if such versions are in use, but a partial version returns an error.

Chapter 17 Asset Management

This chapter describes the asset pages for Hosts, Agents, and NDB Clusters and describes how to remove asset information from the MySQL Enterprise Monitor inventory.

17.1 Agents View

The Agents view configuration view enables you to view all MySQL Enterprise Monitor Agents currently registered with the MySQL Enterprise Service Manager.

To view the agent status and configuration, select **Agents** from the **Configuration** section of the navigation bar.

The agents are displayed on individual lines, with brief summaries of their current status, number of assets monitored by the agent, and last contact.

Figure 17.1 Agents View

Agents		All Agents	Configuration View
Select All	Delete	Total: 80 Agents	
	Agent01 Version: 8.0.13.1154	1 Monitored Assets	MySQL01:3306 Version: 5.7.24-log Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:44 PM Data Dir: /var/lib/mysql/
	Agent02 Version: 8.0.13.1154	1 Monitored Assets	MySQL02:3306 Version: 5.7.24-ndb-7.6.8-cluster-gpl Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:34 PM Data Dir: /var/lib/mysql/
	Agent03 Version: 8.0.13.1154	0 Monitored Assets	-
	Agent04 Version: 8.0.13.1154	0 Monitored Assets	-
	Agent05 Version: 8.0.13.1154	1 Monitored Assets	MySQL05:3306 Version: 5.7.24-log Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:41 PM Data Dir: /var/lib/mysql/
	Agent06 Version: 8.0.13.1154	1 Monitored Assets	MySQL06:3306 Version: 5.7.24-log Instance Uptime: a day Last Contact: Nov 23, 2018 12:26:29 PM Data Dir: /var/lib/mysql/

To view detailed information on the agent's configuration, select **Details** from the more info menu on the right-hand side of each line.



Note

It is also possible to delete agents from the MySQL Enterprise Monitor inventory using the **Delete** option. For more information on deleting assets, see [Section 17.4, "Deleting Assets"](#).

Figure 17.2 Agent Details View

List > Agent01

Agent: Agent01

Host Name: Host01
 Home Directory: /home/mysql/agent/bin/..
 Id: b186b8b0-96a7-e4fb-6a6b-ebc8d6817312
 Version: 8.0.13.1154
AGENT REACHABLE

JVM MySQL Connections MySQL Processes MySQL Instances OS

```

Awt Graphicsenv: sun.awt.X11GraphicsEnvironment
Awt Printerjob: sun.print.PSPrinterJob
Class Path: /home/mysql/agent/bin/bootstrap
Class Version: 52.0
Endorsed Dirs: /home/mysql/agent/java/lib/endorsed
Ext Dirs: /home/mysql/agent/java/lib/ext:/usr/java/packages/lib/ext
Home: /home/mysql/agent/java
Io Tmpdir: /tmp
Library Path: /home/mysql/agent/bin/./lib
Runtime Version: 1.8.0_192-b12
Specification Name: Java Platform API Specification
Specification Vendor: Oracle Corporation
Specification Version: 1.8
Vendor: Oracle Corporation
Vendor Url: http://java.oracle.com/
Version: 1.8.0_192
VM Info: mixed mode
VM Name: Java HotSpot(TM) Server VM
VM Specification Name: Java Virtual Machine Specification
VM Specification Vendor: Oracle Corporation
VM Specification Version: 1.8
VM Vendor: Oracle Corporation
VM Version: 25.192-b12

```

The agent details page is divided into the following tabs:

- **JVM**: displays the details of the agent's Java Virtual Machine (JVM).
- **MySQL Connections**: (displayed only if the agent is monitoring an instance) lists all the MySQL connections currently monitored by the agent and their configuration details. Information such as whether self-signed certificates are permitted, timeout values, hostnames, and so on.
- **MySQL Processes**: (displayed only if the agent is monitoring an instance) lists all the monitored MySQL processes. Information such as the arguments with which the process was started, process ID, and so on.
- **MySQL Instances**: (displayed only if the agent is monitoring an instance) lists all the monitored MySQL instances. Information such as the version, status, uptime, and so on.
- **OS**: (displayed only if the agent is configured to monitor the host) : displays detailed information the host operating system, file systems, CPUs, and so on.

17.2 Hosts View

The **Hosts** configuration view enables you to view all currently monitored hosts and their configuration.

To view the monitored hosts, select **Hosts** from the **Configuration** section of the navigation bar.

The hosts are displayed on individual lines, with brief summaries of their current status, number of instances installed, and information on their operating systems and hardware.

Figure 17.3 Hosts View

Hosts		All Hosts	Configuration View
Select All	X Delete	Total: 80 Hosts	
	Host01 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores) Total RAM: 251.77 GiB Total Swap: 4 GiB
	Host02 OS: Oracle Solaris 11 (sparc) Version: 11	MySQL Instances 0	Filesystems: 30 Networks: 2 CPUs: 16 (16 cores) Total RAM: 32 GiB Total Swap: 8 GiB
	Host03 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores) Total RAM: 251.77 GiB Total Swap: 4 GiB
	Host04 OS: Oracle Linux 7.2 (x86_64) Version: 7.2	MySQL Instances 0	Filesystems: 96 Networks: 18 CPUs: 88 (88 cores) Total RAM: 503.78 GiB Total Swap: 4 GiB
	Host05 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores) Total RAM: 251.77 GiB Total Swap: 4 GiB
	Host06 OS: Oracle Linux 6.5 (x86_64) Version: 6.5	MySQL Instances 1	Filesystems: 15 Networks: 2 CPUs: 2 (2 cores) Total RAM: 7.55 GiB Total Swap: 4 GiB
	Host07 OS: Oracle Linux 7.5 (x86_64) Version: 7.5	MySQL Instances 1	Filesystems: 40 Networks: 2 CPUs: 40 (40 cores) Total RAM: 251.77 GiB Total Swap: 4 GiB

The host names are displayed on individual lines, with brief summaries of their membership, current status.

To view detailed information on the host's configuration, filesystems, and so on, select **Details** from the more info menu on the right-hand side of each line.



Note

It is also possible to delete hosts, using the **Delete** option. For more information on deleting assets, see [Section 17.4, "Deleting Assets"](#).

Figure 17.4 Host Details View

List > {os.Os : Host01}

Host: {os.Os : Host01}

```

Architecture: x86_64
Description: Oracle Linux 7.5
Machine: x86_64
Name: Linux
Patchlevel: unknown
Vendor: Oracle
Vendor Code Name:
Vendor Name: Linux
Vendor Version: 7.5
Version: 4.14.35-1818.0.9.el7uek.x86_64
    
```

Type	System Type	Total Bytes	Used Bytes	Unused Bytes	Available Bytes	Dev Name	Dir Name
none	tmpfs	64 MIB		64 MIB	64 MIB	tmpfs	/proc/keys
local	ext4	365.64 GiB	105.15 GiB	260.49 GiB	241.89 GiB	/dev/sdc1	/home/mysql/agent/etc/agentMi
none	tmpfs	125.88 GiB		125.88 GiB	125.88 GiB	tmpfs	/sys/firmware
none	mqueue					mqueue	/dev/mqueue
none	cgroup					cgroup	/sys/fs/cgroup/net_cls,net_prio
none	cgroup					cgroup	/sys/fs/cgroup/cpu,cpuacct
local	ext4	365.64 GiB	105.15 GiB	260.49 GiB	241.89 GiB	/dev/sdc1	/home/mysql/agent/spool
none	cgroup					cgroup	/sys/fs/cgroup/blkio
local	ext4	365.64 GiB	105.15 GiB	260.49 GiB	241.89 GiB	/dev/sdc1	/etc/resolv.conf
none	tmpfs	64 MIB		64 MIB	64 MIB	tmpfs	/proc/kcore
none	cgroup					cgroup	/sys/fs/cgroup/hugetlb

The Hosts details view contains the following tabs:

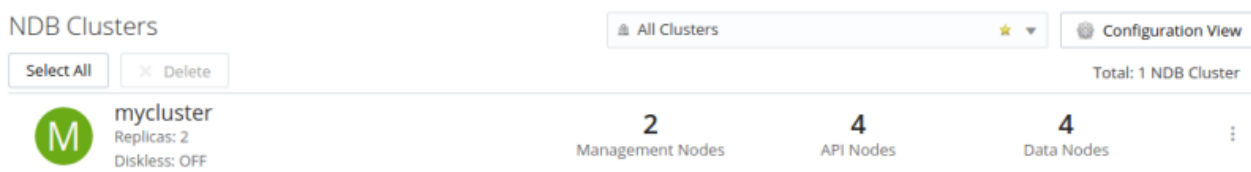
- **Filesystems:** lists all the file systems available to the host. Information includes the bytes used, bytes unused, file system type, and so on.
- **Memory:** lists the total amount of RAM available and the size of the swap file.
- **CPUs:** lists the CPUs installed on the host. Information includes number of cores, cache size, frequency, and so on.
- **MySQL Instances:** lists all the monitored MySQL instances. Information such as the version, status, uptime, and so on.
- **Networks:** lists all the network interfaces on the host, their types, names, and so on.

17.3 NDB Clusters View

The **NDB Clusters** configuration view enables you to view all currently monitored clusters, their membership, and configuration.

To view the monitored cluster configuration and status, select **NDB Clusters** from the **Configuration** section of the navigation bar.

Figure 17.5 NDB Clusters View



The cluster names are displayed on individual lines, with brief summaries of their membership, current status.

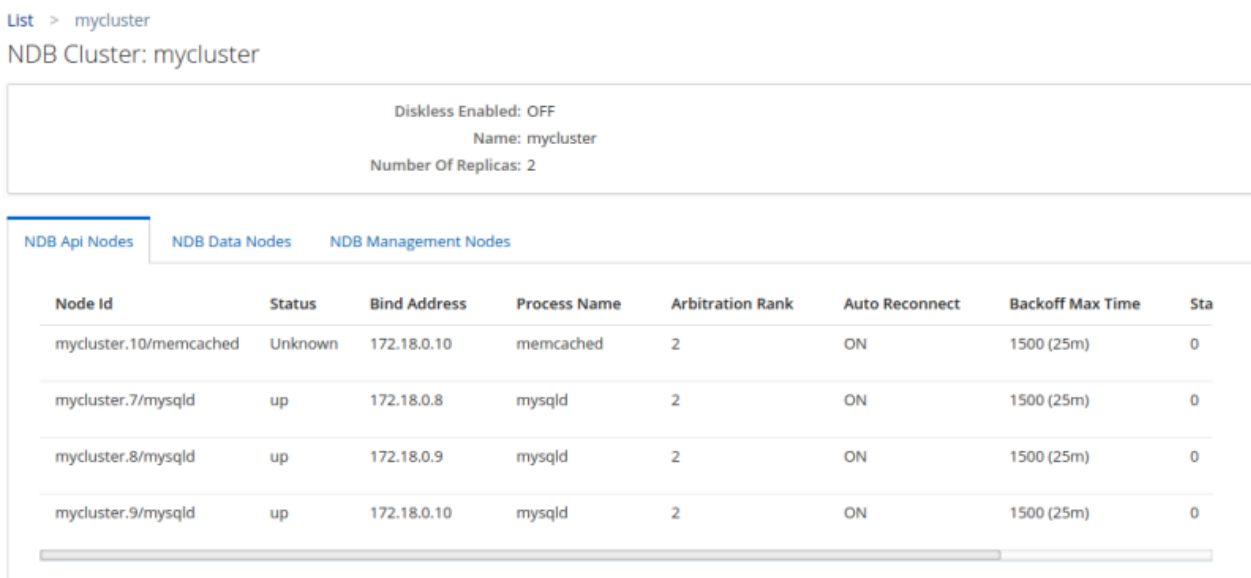
To view detailed information on the cluster's membership, select **Details** from the more info menu on the right-hand side of each line.



Note

It is also possible to delete clusters, or members of the cluster, using the **Delete** option. For more information on deleting assets, see [Section 17.4, “Deleting Assets”](#).

Figure 17.6 NDB Clusters View



The cluster details view displays a summary of the cluster's members organized by node type. Each tab displays the current state of each node and pertinent configuration information, such as **Bind Address**, **Process Name**, **Auto Reconnect**, and so on.

To drill down into the details of each node's configuration, select **Details** from the more info menu on the right-hand side of each line.

The node details view displays all available configuration and state information for each monitored node.

17.4 Deleting Assets

This section describes how to delete assets, and groups of assets, from MySQL Enterprise Monitor's inventory, from the user interface.



Important

Deleting assets from the inventory does not delete or uninstall assets from their installed locations. The delete process removes all record of an asset or group of assets from the internal record of MySQL Enterprise Service Manager, only. This enables you to remove entries for clusters, topologies, agents, hosts, and instances which were monitored, but are no longer in use.

Ensure the asset or group is either not running, or not installed, before attempting to delete it from the inventory.

For more information on the inventory, see [Appendix B, *Managing the Inventory*](#).

For information on deleting assets from the command line, see [Section 17.5, “Deleting Assets from the Command Line”](#).

The deletable asset types are as follows:

- MySQL Instances
- MySQL Enterprise Monitor Agents
- Hosts
- NDB Clusters: Clusters can be deleted in their entirety, or you can delete individual nodes by opening the NDB Cluster details view, and deleting the nodes one at a time.
- NDB API Nodes



Note

Deleting an NDB API Node also deletes the server.

- NDB Data Nodes
- NDB Management Nodes



Note

It is not currently possible to delete an entire replication topology. To delete the assets involved in a replication topology, you must delete each individually, or use a related request type where possible.

It is not possible to delete an asset if any of the following are true:

- If the user does not have the required permissions.
- It is not possible to delete an agent from the inventory if the agent's status is unknown, if the agent is running, or if the agent is monitoring one or more MySQL instances which are not scheduled for deletion in the same operation.
- It is not possible to delete a host from the inventory if the host contains agents or MySQL instances which are not also scheduled for deletion in the same operation.

To delete an asset, do the following:

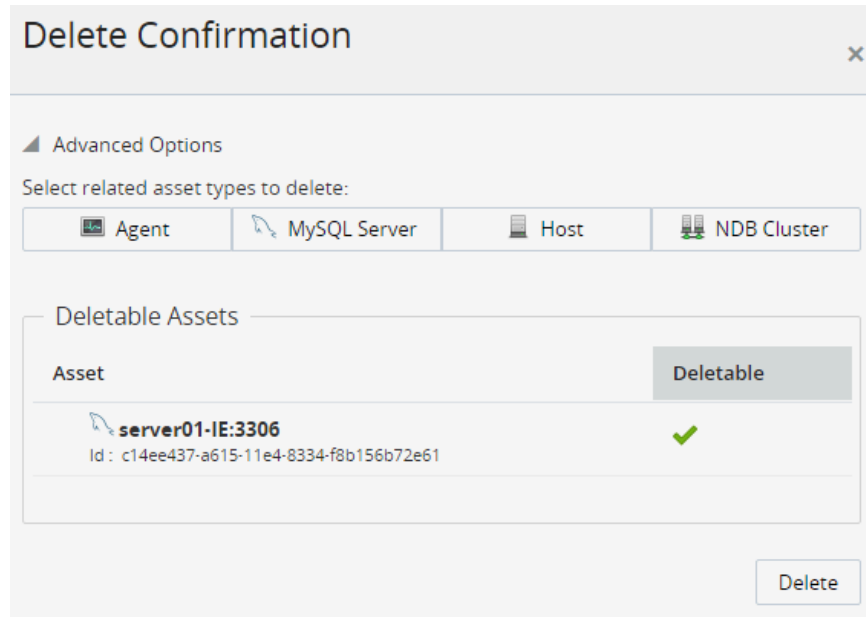
1. Open the asset's view from the **Configuration** menu.

- To delete an MySQL instance from the **MySQL Instances** view, select **Delete** from the instance menu, or select the instance(s) and click the **Delete Instances** button.

To delete an asset from the asset-specific views, Agents, Hosts, or NDB Clusters, select **Delete** from the menu on each asset's line or select the asset and click the Delete button. You can also select all assets in the view, using the **Select All** button, and delete all by clicking the **Delete** button.

The **Delete Confirmation** dialog is displayed.

Figure 17.7 Delete Confirmation for MySQL Instance



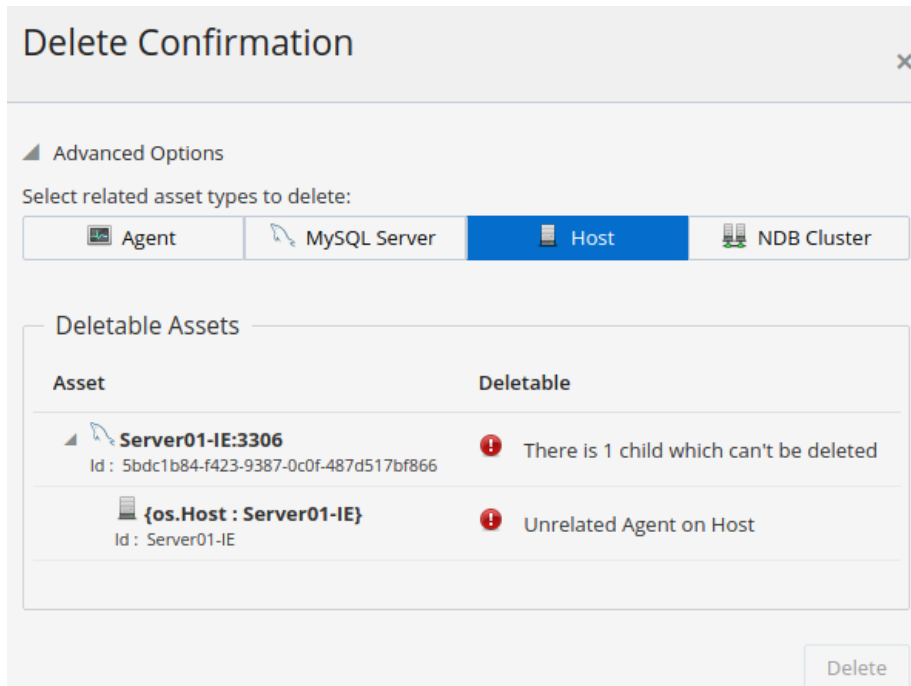
If it is not possible to delete an asset, a warning is displayed in the **Deletable** column. To see more information on the nature of the warning, hover the cursor over the warning. A tooltip is displayed.

- If the asset is deletable, confirm the details are correct and click **Delete**. A confirmation dialog is displayed.

- To delete assets related to the selected asset, select one of the other types from the **Advanced Options** section. If the selection is available, the **Deletable Assets** section is updated with the asset's details and whether it is deletable.

For example, if you try to delete a MySQL Instance and a related host:

Figure 17.8 Delete Confirmation for MySQL Instance and related Host



The related host also has an agent installed, which is unrelated to the delete request. The presence of the unrelated agent prevents the removal of the MySQL instance and related host delete request. The **Delete** button is deactivated.

17.5 Deleting Assets from the Command Line

This section describes how to delete information on assets, and groups of assets, from MySQL Enterprise Monitor's inventory, using the `curl` utility on the command line.



Important

Deleting assets from the inventory does not delete or uninstall assets from their installed locations. The delete process removes all record of an asset or group of assets from the internal record of MySQL Enterprise Service Manager, only. This enables you to remove entries for clusters, topologies, and instances which were monitored, but are no longer in use.

Ensure the asset or group is either not running, or not installed, before attempting to delete it from the inventory.

For more information on the inventory, see [Appendix B, Managing the Inventory](#).

This section describes the following:

- [Retrieving Deletable Asset Information](#): how to check what is deletable.

- [Listing the Related Deletable Assets](#): how to check for related assets and their status.
- [Deleting Assets from the Inventory](#): how to delete assets.
- [Request Status Codes](#): return codes from the various request types.

Retrieving Deletable Asset Information

The curl utility can be used to request information directly from MySQL Enterprise Service Manager's inventory of monitored assets. To retrieve information on whether the specific asset can be deleted, retrieved as a JSON array, execute a curl request with the following syntax:

```
curl -k -v --user username:password
      "https://servername:port/v4/inventory/AssetType/AssetID/deletable"
```

where:

- `username:password` is the username and password of a user with admin access to MySQL Enterprise Service Manager.
- `servername:port` is the servername and port number for MySQL Enterprise Service Manager.
- `AssetType` is the type of asset information you want to retrieve. The valid asset types are as follows:
 - MySQL Instances: `com.mysql.ertools.inventory.model.mysql.MysqlServer`
 - MySQL Enterprise Monitor Agents: `com.mysql.ertools.inventory.model.agent.Agent`



Note

It is not possible to delete an agent from the inventory if the agent's status is unknown, if the agent is running, or if the agent is monitoring one or more MySQL instances which are not scheduled for deletion in the same operation.

- Hosts: `com.mysql.ertools.inventory.model.os.Host`



Note

It is not possible to delete a host from the inventory if the host contains agents or MySQL instances which are not also scheduled for deletion in the same operation.

- NDB Clusters: `com.mysql.ertools.inventory.model.mysql.cluster.NdbCluster`
- NDB API Nodes: `com.mysql.ertools.inventory.model.mysql.cluster.NdbApiNode`



Note

Deleting an NDBAPINode also deletes the server.

- NDB Data Nodes: `com.mysql.ertools.inventory.model.mysql.cluster.NdbDataNode`
- NDB Management Nodes: `com.mysql.ertools.inventory.model.mysql.cluster.NdbManagementNode`
- `AssetID` is the unique, alphanumeric identifier of the asset type.

**Note**

To find the [AssetID](#), check the inventory page for the specific asset type. See [Appendix B, *Managing the Inventory*](#) for more information.

The following example retrieves the information for a MySQL instance with the unique identifier `c14ee437-a615-11e4-8334-f8b156b72e61`:

**Note**

In the following example, line breaks have been added for clarity.

```
curl -k -v --user admin:mysql
https://localhost:18443/v4/inventory/com.mysql.ertools.inventory.model.mysql.MySqlServer/
c14ee437-a615-11e4-8334-f8b156b72e61/deletable"
```

If the MySQL instance is deletable, the request returns the following data, confirming that the asset type can be deleted from the inventory:

```
"asset" : {
  "id" : "c14ee437-a615-11e4-8334-f8b156b72e61",
  "assetClass" : "com.mysql.ertools.inventory.model.mysql.MySqlServer"
},
"isDeletable" : true,
"treeDeletable" : true,
"problems" : [ ],
"token" : "3b66cd39467791a739e5131a8d6a856b"
```

where:

- `asset`: lists the id and asset class of the requested asset.
- `isDeletable`: True|False. Lists whether the asset's information can be deleted from the inventory.
- `treeDeletable`: True|False. Lists whether related assets can be deleted.
- `problems`: lists the reasons why the asset's information cannot be deleted from the inventory. Possible reasons are:
 - `ACL_FAIL`: the user has insufficient privileges to perform the operation.

Deleting an asset requires the **Administer** permission for **Server Group and MySQL NDB Cluster**. To delete an asset from the inventory, the user must belong to a role with that permission.

For more information, see [Section 23.3, "Monitored Assets Permissions"](#).

- `IS_LIVE`: the agent is running. The agent is running and broadcasting information to the Service Manager. It is not possible to delete a running asset from the inventory. To delete an asset from the inventory, you must first stop or uninstall it.
- `UNKNOWN_LIVENESS`: The agent's state could not be determined. This problem can be caused by network issues, or some other issue, such as a power cut, preventing the Service Manager from contacting the agent.

- `AGENT_HAS_UNRELATED_SERVER`: it is not possible to delete a server and related agent if the agent is monitoring a MySQL instance not included in the scope of the request. To include it, you must broaden the scope of the related request by including the `agent` keyword.
- `OS_HAS_UNRELATED_AGENT`: it is not possible to delete a host which contains an agent not scheduled for deletion. The host contains an agent which is not included in the request's scope. To include it, you must broaden the scope of the related request by including the `agent` keyword.
- `OS_HAS_UNRELATED_SERVER`: it is not possible to delete a host which contains a MySQL instance not scheduled for deletion. The host contains a MySQL instance which is not included in the request's scope. To include it, you must broaden the scope of the related request by including the `mysqlserver` keyword.
- `token`: a unique alphanumeric identifier for the state of the asset. This token can be used in a `DELETE` request, to confirm the state of the asset has not changed between the time the `deletable` request was run, and the time the `DELETE` request is run.

The request also returns a status code indicating the success or failure of the request. For more information, see [Request Status Codes](#).

Listing the Related Deletable Assets

The `related` request enables you to return data on other assets which are related to the primary asset defined in the `type` field. For example, using `related`, you can request deletable information for a MySQL instance and for the agent monitoring it, the host on which it resides, and so on. `related` also enables you to delete multiple assets from the inventory with one request.

`related` takes a comma-separated list of the following values:

- `agent`: any agents related to the primary asset.
- `mysqlserver`: any MySQL instances related to the primary asset.
- `ndbcluster`: any NDBClusters related to the primary asset.
- `os`: any hosts related to the primary asset.

The syntax of the request is as follows:

```
curl -k -v --user username:password
"https://servername:port/v4/inventory/AssetType/AssetID/deletable?
related=agent,mysqlserver,ndbcluster,os"
```

For example, the following requests deletable information on a MySQL instance with the AssetID `c14ee437-a615-11e4-8334-f8b156b72e61` and on the agent monitoring it:

```
curl -k -v --user admin:mysql
"https://localhost:18443/v4/inventory/com.mysql.ertools.inventory.model.mysql.MysqlServer/
c14ee437-a615-11e4-8334-f8b156b72e61/deletable?related=agent"
```

The following JSON array is returned:

```
"asset" : {
  "id" : "c14ee437-a615-11e4-8334-f8b156b72e61",
  "assetClass" : "com.mysql.ertools.inventory.model.mysql.MySqlServer"
},
"isDeletable" : true,
"treeDeletable" : false,
"problems" : [ ],
"children" : [ {
  "asset" : {
    "id" : "b68947f9-39f2-4003-981f-8c9cc7a7409f",
    "assetClass" : "com.mysql.ertools.inventory.model.agent.Agent"
  },
  "isDeletable" : false,
  "treeDeletable" : false,
  "problems" : [ {
    "code" : "IS_LIVE",
    "description" : "IS_LIVE"
  } ]
} ],
} ],
"token" : "d0e3607f6029a91a35a9956950908761"
```

The data returned shows that while the MySQL instance is deletable, it is not `treeDeletable`, which means it cannot be deleted in a related `DELETE`, because the monitoring agent is still running.

Deleting Assets from the Inventory

To delete individual assets from MySQL Enterprise Service Manager's inventory, use the following syntax:



Note

Ensure the asset or group of assets is either not running, or not installed, before attempting to delete it from the inventory.

```
curl -k -v -X DELETE --header "X-Requested-With:XMLHttpRequest"
--header "If-Match:tokenID"
--user admin:mysql
"https://servername:port/v4/inventory/AssetType/AssetID"
```

To delete related assets from MySQL Enterprise Service Manager inventory, use the following syntax:

```
curl -k -v -X DELETE --header "X-Requested-With:XMLHttpRequest"
--header "If-Match:tokenID"
--user admin:mysql
"https://servername:port/v4/inventory/AssetType/AssetID?
related=agent,mysqlserver,ndbcluster,os"
```



Note

`--header "If-Match:tokenID"` is optional and should only be used if you want to confirm the deletable information is still accurate. If the tokenID is included, and matches the value retrieved by the deletable request, the `DELETE` operation is performed.

For example, the following shows a deletable request with the return data, listing a deletable agent, MySQL instance, the host it used to monitor, and another agent which was resident on the same server:


```
curl -k -v --user admin:mysql
"https://localhost:18443/v4/inventory/com.mysql.etools.inventory.model.agent.Agent/
d71ff41c-affd-4ebe-966e-395e6f06b5e3/deletable?related=mysqlserver,os,agent"

{
  "asset" : {
    "id" : "d71ff41c-affd-4ebe-966e-395e6f06b5e3",
    "assetClass" : "com.mysql.etools.inventory.model.agent.Agent"
  },
  "isDeletable" : true,
  "treeDeletable" : true,
  "problems" : [ ],
  "children" : [ {
    "asset" : {
      "id" : "sid:{S-1-5-21-700548433-2193620149-503502644}",
      "assetClass" : "com.mysql.etools.inventory.model.os.Host"
    },
    "isDeletable" : true,
    "treeDeletable" : true,
    "problems" : [ ],
    "children" : [ {
      "asset" : {
        "id" : "b68947f9-39f2-4003-981f-8c9cc7a7409f",
        "assetClass" : "com.mysql.etools.inventory.model.agent.Agent"
      },
      "isDeletable" : true,
      "treeDeletable" : true,
      "problems" : [ ],
      "children" : [ {
        "asset" : {
          "id" : "c14ee437-a615-11e4-8334-f8b156b72e61",
          "assetClass" : "com.mysql.etools.inventory.model.mysql.MySQLServer"
        },
        "isDeletable" : true,
        "treeDeletable" : true,
        "problems" : [ ]
      } ]
    } ]
  } ],
  "token" : "7048b6e53c81f549b55c5f3e861c2eee"
```

The following example shows a related DELETE request, deleting all assets listed as deletable in the deletable example:

```
curl -k -v -X DELETE --header "X-Requested-With:XMLHttpRequest"
--user admin:mysql "https://localhost:18443/v4/inventory/com.mysql.etools.inventory.model.agent.Agent/
d71ff41c-affd-4ebe-966e-395e6f06b5e3?related=mysqlserver,os,agent"
```



Note

It is not currently possible to delete an entire replication topology, unlike NDB Cluster, which can be deleted using the [com.mysql.etools.inventory.model.mysql.cluster.NdbCluster](#) Asset type.

To delete the assets involved in a replication topology, you must delete each individually, or use a [related](#) request type where possible.

Request Status Codes

The status code can be found in the following section of the return message, following the HTTP/1.1:

```
HTTP/1.1 200
Set-Cookie: JSESSIONID=8C0C2E1AFD45F170E7E41621F934723D; Path=/; Secure; HttpOnly
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Content-Type-Options: nosniff
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 24 Jul 2018 12:31:47 GMT
```

The following status codes are returned for `deletable` and `DELETE` requests:

- **200**: the deletable request was successful. Content is returned in a JSON array.
- **204**: the DELETE request was successful. No content is returned.
- **400**: the request is not valid. Check the asset information, or the syntax of the request for errors, typos, and so on.
- **403**: (`DELETE` request, only) the `X-Requested-With` header is not present.
- **404**: incorrect AssetID provided.
- **409**: it was not possible to perform the DELETE request. Further information is provided in a JSON array.
- **412**: the requested token check failed.

Chapter 18 Managing Groups of Instances

Groups organize 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.

There are two types of group in MySQL Enterprise Monitor: user-defined and replication. User-defined groups are those created and managed in MySQL Enterprise Service Manager's **Manage Groups** page. An instance can belong to one or more groups.



Note

You can also define groups and add instances to those groups using the MySQL Enterprise Monitor Agent configuration utility. For information, see [Agent Connection Utilities](#). You can use the configuration utility to add connections to existing groups, or to create a group and add a connection to it, but cannot delete an existing group.

Replication groups are instances configured in replication topologies. These groups cannot be managed by MySQL Enterprise Service Manager. 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.



Important

It is not possible to edit replication group membership. Such groups are populated dynamically. The selection boxes are grayed out in Replication groups. It is possible to change the **Group Name** and **Group Description**, only.



Note

NDB Cluster and InnoDB cluster topologies are not manageable and are not displayed in the **Manage Groups** page.

The primary uses for groups are:

- **Access Control:** You can assign users to specific groups. 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](#).
- **Organization:** grouping related instances together, in order to ensure consistent Advisor scheduling and event generation. An instance can belong to multiple groups. It is not possible for instances involved in a replication topology to belong to multiple groups.

Creating Groups

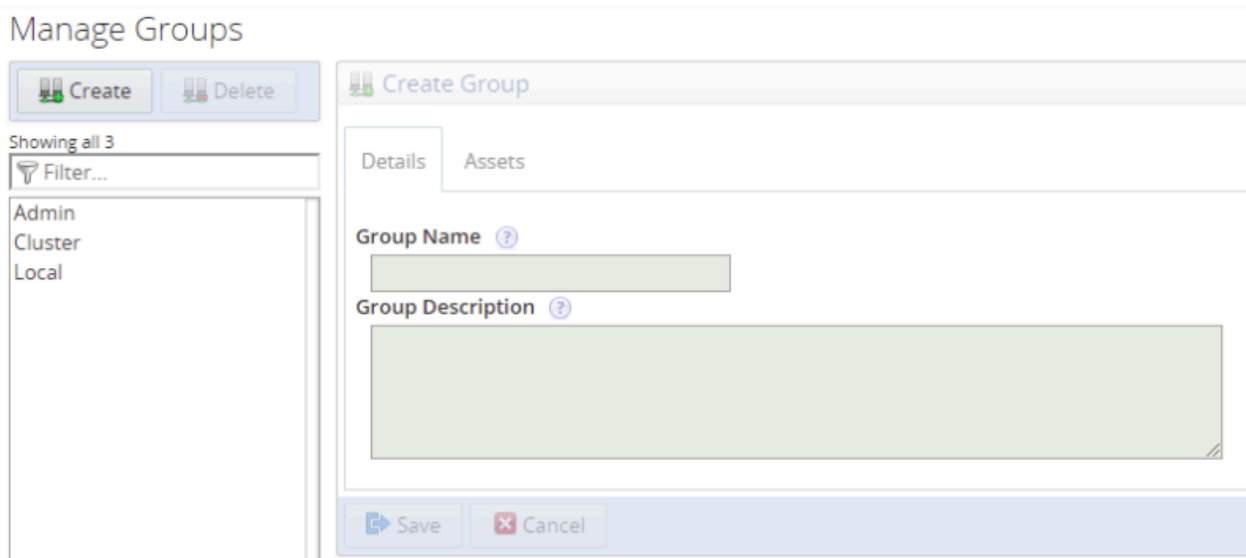


Important

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.

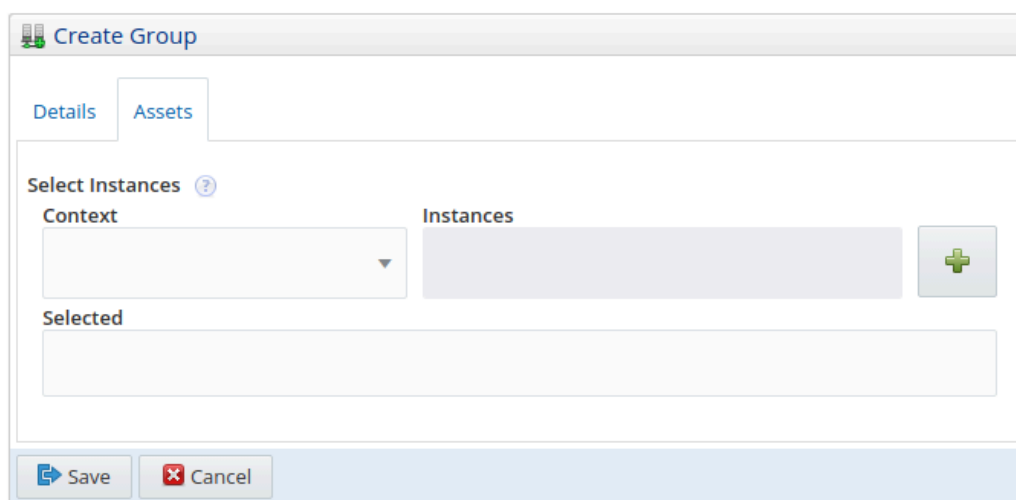
Figure 18.1 Group Management Page



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.

Figure 18.2 Group Assets Tab



5. Select the required instances by selecting a context. It is not possible to select instances without selecting a context.
6. Select **All** to make all instances available for selection in the **Instances** field. To filter further, select any of the existing groups to make their contents available in the **Instances** field.

7. Select an instance by clicking in the Instances field and selecting the required instances. This field also supports auto-complete.
8. Click the add button to add your selection to the **Selected** field.
9. Click **Save** to save your new group. Click Cancel to discard your changes.
10. 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 19 Advisors

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:

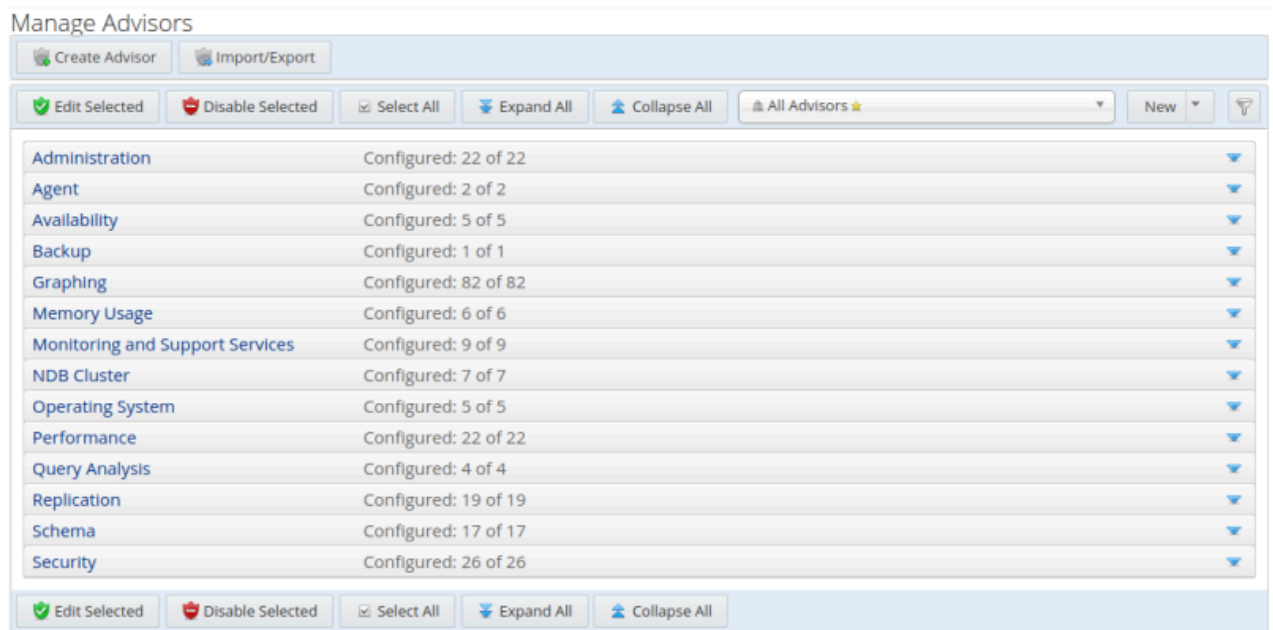
- [Section 19.1, “Manage Advisors Page”](#)
- [Section 19.2, “Advisor Types”](#)
- [Section 19.3, “Advisor Thresholds”](#)
- [Section 19.4, “Advisor Schedules”](#)

19.1 Manage Advisors Page

This section describes the main Advisors page.

To display the Advisors page, select **Advisors** from the **Configuration** menu.


Figure 19.1 Manage Advisors Page



The components and controls of the **Manage Advisors** page are as follows:

Table 19.1 Manage Advisors Page Controls

Name	Description
Edit Selected	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

Name	Description
	Advisors simultaneously. You can also edit an advisor using the drop-down menu adjacent to each advisor's name.
Disable Selected	Disables all selected Advisors.
Create Advisor	Opens the Create Advisor page.
Import/Export	Opens the Custom Rule/Graph Export page. 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. For information on the named filters, see Section 14.3.5, "System and User-defined Filters" .

Advisor Categories

The following types of Advisor are provided:

- **Administration:** Checks the MySQL instance installation and configuration.
- **Agent:** Checks the status of each MySQL Enterprise Monitor Agent.
- **Availability:** Checks the availability of the MySQL process and the connection load.
- **Backup:** Checks whether backup jobs succeed or fail, required resources, and information about MySQL Enterprise Backup specific tasks.
- **Cluster:** Checks the status of the monitored MySQL Cluster.
- **Graphing:** Data for graphs.
- **Memory Usage:** Indicate how efficiently you are using various memory caches, such as the InnoDB buffer pool, MyISAM key cache, query cache, table cache, and thread cache.
- **Monitoring and Support Services:** Advisors related to the MySQL Enterprise Monitoring services itself.
- **Operating System:** Checks the Host Operating System performance.
- **Performance:** Identifies potential performance bottlenecks, and suggests optimizations.
- **Query Analysis:** Advisors related to Queries and Query Analysis.
- **Replication:** Identifies replication bottlenecks, and suggests replication design improvements.
- **Schema:** Identifies schema changes.
- **Security:** Checks MySQL Servers for known security issues.

It is also possible to create custom Advisors.

To display the Advisors in each category, click on the Category name. For a full description of the default advisors, see [Chapter 21, GUI-Based Advisor Reference](#) and [Chapter 20, Expression-Based Advisor Reference](#).

Advisors configure the type of data collected by the Agent. If you do not want to monitor for a specific type of data, disabling the Advisor responsible for that data type instructs the Agents to stop collecting that data.

Advisor Listing Table

The listing table displays all categories, Advisors, monitored instances, and displays information on the configuration of the Advisors.

The configuration information is displayed in the following columns:

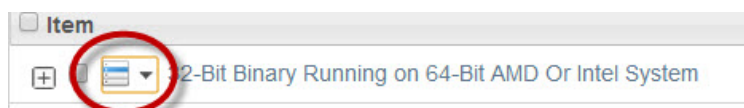
Table 19.2 Advisor Information Listing

Name	Description
Item	Displays the Advisor name, group name, and monitored instance name. To expand the Advisor, click the expand icon.
Info	Click to display a tooltip which describes the Advisor.
Coverage	Displays the Advisor's coverage of the monitored instance. If the Advisor has been edited for a specific instance, this field is empty for that instance. If the default Advisor settings are used, this field displays (Covered) .
Schedule	Displays the defined evaluation schedule. If the Advisor is disabled, this field displays Disabled for the level at which it was disabled, Advisor, Group or monitored instance.
Event Handling	Displays the event handling status icons. For more information, see Chapter 22, Event Handlers .
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

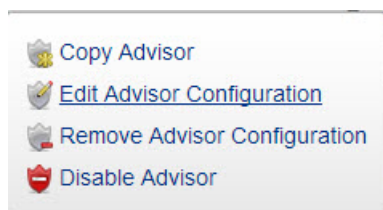




Table 19.3 Advisor Edit Menu Controls

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.

Name	Description
	 <p>Note</p> <p>This option is only available for expression-based Advisors.</p>
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	<p>Deletes the selected advisor.</p>  <p>Note</p> <p>Only available for custom Advisors. It is not possible to delete the default Advisors.</p>

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. For information on the named filters, see [Section 14.3.5, "System and User-defined Filters"](#).

The Advisor Filter is displayed:

Figure 19.4 Advisor Filter Controls

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: <ul style="list-style-type: none"> • 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	Only available for named filters, not for the system filters. Filters the Advisors list based on the search terms and expands the categories and Advisors to display the search results.
Discard Changes	Removes any changes made to the saved filter.
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.

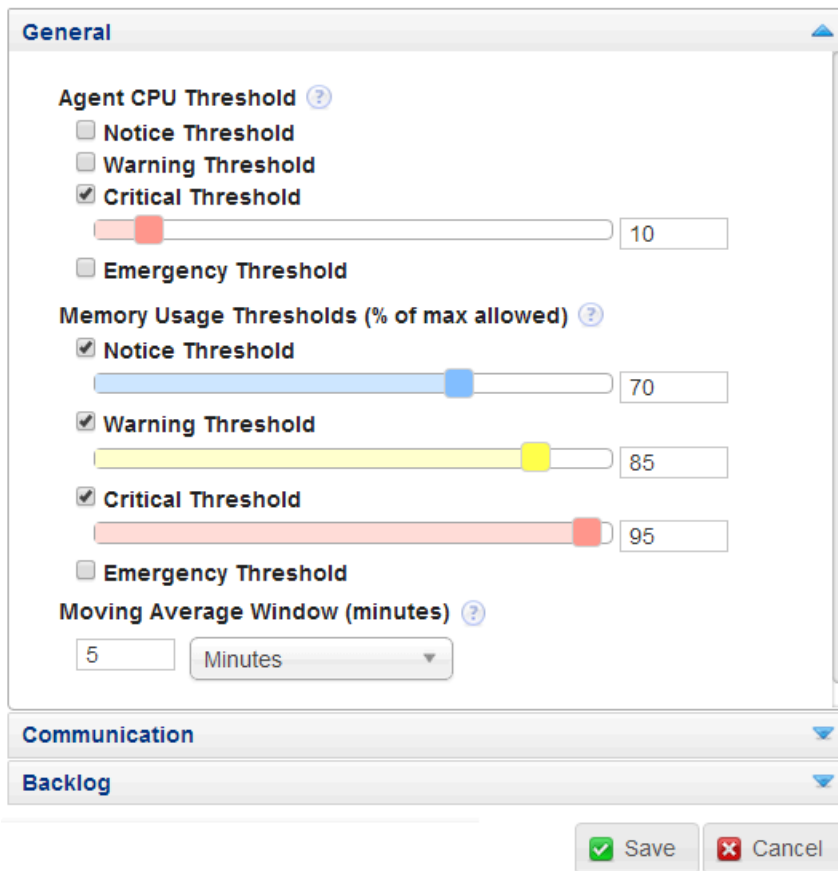
Expression-based advisors are described in [Chapter 20, Expression-Based Advisor Reference](#).

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



GUI-based advisors are described in [Chapter 21, GUI-Based Advisor Reference](#).

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 can soon affect the performance of the server. Such issues require immediate attention.
- **Emergency:** indicates a serious problem with the server. The server is unusable or unresponsive and requires immediate attention.

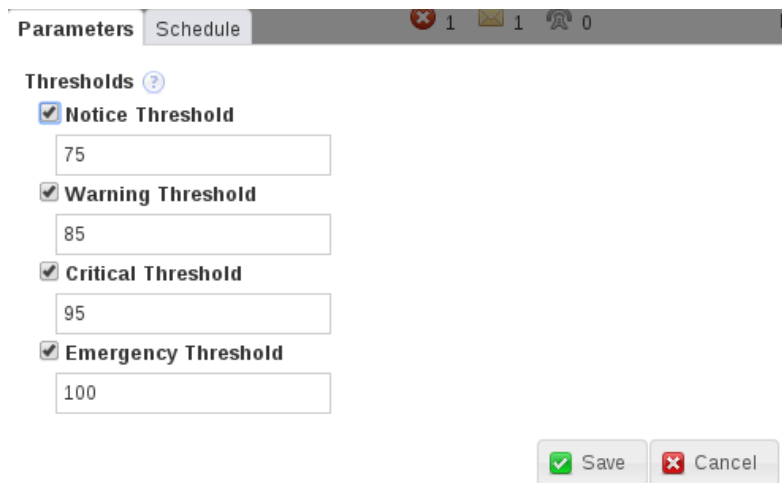


Note

Not all Advisors require threshold parameters, others do not have any parameters, such as the **Graphing** Advisors.

The following image shows an example of threshold definitions on the Parameters tab of an advisor:

Figure 19.6 Threshold Definitions Example



The 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 Expression-Based Advisor Reference

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

20.1 Administration Advisors

This section describes the expression-based Administration Advisors.

- [32-Bit Binary Running on 64-Bit AMD Or Intel System](#)
- [Binary Log Debug Information Disabled](#)
- [Binary Logging Is Limited](#)
- [Binary Logging Not Enabled](#)
- [Binary Logging Not Synchronized To Disk At Each Write](#)
- [Binary Logs Automatically Removed Too Quickly](#)
- [Database May Not Be Portable Due To Identifier Case Sensitivity](#)
- [Event Scheduler Disabled](#)
- [General Query Log Enabled](#)
- [Host Cache Size Not Sufficient](#)
- [In-Memory Temporary Table Size Limited By Maximum Heap Table Size](#)
- [InnoDB Status Truncation Detected](#)
- [InnoDB Strict Mode Is Off](#)
- [InnoDB Tablespace Cannot Automatically Expand](#)
- [InnoDB Transaction Logs Not Sized Correctly](#)
- [Multiple Threads Used When Repairing MyISAM Tables](#)
- [MySQL Server No Longer Eligible For Oracle Premier Support](#)
- [Next-Key Locking Disabled For InnoDB But Binary Logging Enabled](#)
- [No Value Set For MyISAM Recover Options](#)
- [Table Cache Set Too Low For Startup](#)
- [Time Zone Data Not Loaded](#)
- [Warnings Not Being Logged](#)

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

Raises an event if a 32-bit binary is detected running on a 64-bit platform. Most 32-bit binaries can run on a 64-bit platform. However, for performance reasons, it is recommended to run 64-bit binaries on 64-bit platforms, and 32-bit binaries on 32-bit platforms.

Default frequency 06:00:00

Default auto-close enabled yes

Binary 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 `binlog_rows_query_log_events` to `TRUE`. By default, this advisor generates an event if ROW or MIXED logging is enabled and `binlog_rows_query_log_events=FALSE`.



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 file system case sensitivity.

Default frequency 06:00:00

Default auto-close enabled yes

Event Scheduler Disabled

The Event Scheduler is a framework for executing SQL commands at specific times or at regular intervals, similarly to the Unix crontab or the Windows Task Scheduler.

An event is a stored routine with a starting date and time, and a recurring tag. Unlike triggers, events are not linked to specific table operations, but to dates and times. 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 quickly 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 temp directory. For performance reasons it is recommended to have most temporary tables created in memory, and only create 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 start up 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. MySQL Enterprise Monitor Agent reads this file automatically, if it exists, before executing the `SHOW` statement.

For more information, see [SHOW ENGINE Statement](#).

Default frequency 00:05:00

Default auto-close enabled no

InnoDB Strict Mode Is Off

To guard against ignored typos and syntax errors in SQL, or other unintended consequences of various combinations of operational modes and SQL commands, InnoDB provides a strict mode of operations. In this mode, InnoDB raises error conditions in certain cases, rather than issue a warning and process the specified command. This is analogous to `sql_mode`, which controls what SQL syntax MySQL accepts, and determines whether it silently ignores errors, or validates input syntax and data values.

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

Default frequency 12:00:00

Default auto-close enabled yes

InnoDB Tablespace Cannot Automatically Expand

If the InnoDB tablespace is not allowed to automatically grow to meet incoming data demands and your application generates more data than there is room for, out-of-space errors occur and your application may experience problems.

Default frequency 06:00:00

Default auto-close enabled yes

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



Note

This advisor does not evaluate against MySQL 8, or higher.

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` system variable (the `myisam-recover` option 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 is be `Marked as crashed`, if it becomes corrupt, and no sessions can `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 works 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 date-time 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` do not work.

Default frequency 12:00:00

Default auto-close enabled yes

Warnings Not Being Logged



Note

This advisor does not evaluate against MySQL 8, or higher.

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 do 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

20.2 Agent Advisors

This section describes the expression-based Agent Advisors.

- [MySQL Agent Memory Usage Excessive](#)
- [MySQL Agent Not Reachable](#)

MySQL Agent Memory Usage Excessive

The memory needed by the MySQL Agent for basic monitoring is fairly small and consistent, and depends on the number of rules you have enabled. However, when the Query Analyzer is enabled, the Agent can use significantly more memory to monitor and analyze whatever queries you direct through it. In this case, the amount of memory used depends on the number of unique normalized queries, example queries and example explains being processed, plus the network bandwidth required to send query data to the Service Manager. In general, the amount of memory used for the Query Analyzer is small and well-bounded, but under some circumstances it can become excessive, especially on older versions of Linux.

Default frequency 00:01:00

Default auto-close enabled no

MySQL Agent Not Reachable

In order to monitor a MySQL server, a Service Agent must be running and communicating with the Service Manager. If the Agent cannot communicate with the Service Manager, the Service Manager has no way of knowing if the MySQL database server being monitored is running, and it cannot collect current statistics to properly evaluate the rules scheduled against that server.

Default frequency 00:00:01

Default auto-close enabled yes

20.3 Availability Advisors

This section describes the expression-based Availability Advisors.

- [Attempted Connections To The Server Have Failed](#)
- [Excessive Percentage Of Attempted Connections To The Server Have Failed](#)
- [Maximum Connection Limit Nearing Or Reached](#)
- [MySQL Availability](#)
- [MySQL Server Has Been Restarted](#)

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

20.4 NDB Cluster Advisors

This section describes the expression-based Cluster Advisors.

- [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 DiskPageBuffer Hit Ratio Is Low](#)

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 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.

For more information on [DiskPageBuffer](#), see [The ndbinfo diskpagebuffer Table](#).

Default frequency 00:05:00

Default auto-close enabled no

20.5 Memory Usage Advisors

This section describes the expression-based Memory Usage Advisors.

- [InnoDB Buffer Cache Has Sub-Optimal Hit Rate](#)
- [Key Buffer Size May Not Be Optimal For Key Cache](#)
- [Query Cache Has Sub-Optimal Hit Rate](#)
- [Query Cache Potentially Undersized](#)
- [Table Cache Not Optimal](#)
- [Thread Cache Size May Not Be Optimal](#)

InnoDB Buffer Cache Has Sub-Optimal Hit Rate

Logical I/O is many times faster than physical I/O, and therefore a DBA should strive to keep physical I/O to a minimum. It is true that logical I/O is not free, and that the DBA should work to keep all I/O to a minimum, but it is best if most data access is performed in memory. When using InnoDB, most data access should occur in memory, 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



Note

This advisor does not evaluate against MySQL 8, or higher.

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 indicate 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



Note

This advisor does not evaluate against MySQL 8, or higher.

When the Query Cache is full, and needs to add more queries to the cache, it makes more room in the cache by freeing the least recently used queries from the cache, and then inserting the new queries. If this is happening often, increase the size of the cache to avoid this 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 of 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_open_cache](#), this may indicate performance problems.

Default frequency 00:05:00

Default auto-close enabled no

Thread Cache Size May Not Be Optimal

Each connection to the MySQL database server runs in its own thread. Thread creation takes time, so rather than killing the thread when a connection is closed, the server can keep the thread in its thread cache and use it for a new connection later.

Default frequency 00:05:00

Default auto-close enabled no

20.6 Monitoring and Support Services Advisors

This section describes the Monitoring and Support Services Advisors.

- [HTTP Server Performance](#)
- [Service Manager Health](#)
- [Support Diagnostics](#)
- [Unsupported Agent Version Advisor](#)

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, set the target selectors to **Global Summaries**, select **MEM Service Manager** from the **All Targets** menu, and select **Timeseries Graphs** from the **Metrics** menu.
- 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.

Unsupported Agent Version Advisor

As MySQL Enterprise Monitor is enhanced and extended over time, it is not possible to maintain backward compatibility between new MySQL Enterprise Service Managers and older MySQL Enterprise Monitor Agents. As such, running mismatched versions can result in unpredictable behavior.

It is strongly recommended that you upgrade your agents when upgrading your MySQL Enterprise Service Manager. MySQL Enterprise Service Manager and MySQL Enterprise Monitor Agent should be the same version.

This advisor detects unsupported agents that try to connect to this service manager.

Default frequency 00:01:00

Default auto-close enabled no

20.7 Operating System Advisors

The **CPU Utilization** and **Filesystem Free Space** Advisors are described in [Chapter 21, 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.

20.8 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.

For more information on the binary log and binary log cache, see [The Binary Log](#).

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 from 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 (for example 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 (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 (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`, and so on. 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



Note

This advisor does not evaluate against MySQL 8, or higher.

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.

For more information, see [Defining the Row Format of a Table](#).

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 `INSERT`s block all `SELECT`s 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.

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



Note

This advisor does not evaluate against MySQL 8, or higher.

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 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 plug-in 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 plug-in. This keeps the number of actively executing threads within the server lower, generally leaving less contention for locks and resources, whilst still maintaining very high connection counts from applications.

Default frequency 00:05:00

Default auto-close enabled no

Too Many Concurrent Queries Running

Too many active queries indicates there is a severe load on the server, and may be a sign of lock contention or unoptimized SQL queries.

Default frequency 00:05:00

Default auto-close enabled no

20.9 Replication Advisors

This section describes the Replication Advisors.

- [Binary Log Checksums Disabled](#)
- [Binary Log File Count Exceeds Specified Limit](#)
- [Binary Log Row Based Images Excessive](#)
- [Binary Log Space Exceeds Specified Limit](#)
- [Master Not Verifying Checksums When Reading From Binary Log](#)
- [Relay Log Space Is Very Large](#)
- [Replica Execution Position Too Far Behind](#)
- [Replication Configuration Advisor](#)
- [Replication Status Advisor](#)
- [Replication Too Far Behind](#)
- [Slave Detection Of Network Outages Too High](#)
- [Slave Has Login Accounts With Inappropriate Privileges](#)
- [Slave Not Configured As Read Only](#)

- [Slave Not Verifying Checksums When Reading From Relay Log](#)
- [Slave Relay Logs Not Automatically Purged](#)
- [Slave SQL Processing Not Multi-Threaded](#)
- [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 `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 sources and replicas 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 replica is less than its source.
- `slave_pending_jobs_size_max` or `slave_max_allowed_packet` are less than the source's `max_allowed_packet`.
- When a source is replicating to a replica that has an older version of the MySQL Server than the master

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

Replication Status Advisor

Monitors slave replication status and alerts when replication has stopped or is compromised in some way, for example one of the slave threads has stopped, displays the last error messages seen, and where possible provides specific advice to fix the errors.

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

Replica Execution Position Too Far Behind

When a replica receives updates from its source, the I/O thread stores the data in local files known as relay logs. The replica's applier thread reads the relay logs and executes the updates they contain. If the position from which the applier thread is reading is far 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 replica may not reflect the latest changes made on the source.

Enables definition of the following thresholds:

- **Same File Position Thresholds:** Alert generated if the applier thread reading position and the I/O thread writing position difference is larger than the configured threshold and are reading/writing to the same file.
- **Different File Position Thresholds:** Alerts are generated if the applier thread reading position is larger than the configured threshold and the I/O thread is writing to a different file.

Default frequency 00:05:00

Default auto-close enabled no

Replication Too Far Behind

If a replica is too far behind the source, results of queries directed to the replica may not reflect the latest changes made on the source.

Default frequency 00:01:00

Default auto-close enabled yes

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 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

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 `slave-parallel-type` for more information.

Default frequency 06:00:00

Default auto-close enabled yes

Slave Without REPLICATION SLAVE Accounts

If the master ever fails, you may want to use one of the slaves as the new master. An account with the `REPLICATION SLAVE` privilege must exist for a server to act as a replication master (so a slave can connect to it), so it's a good idea to create this account on your slaves to prepare it to take over for a master if needed.

Default frequency 06:00:00

Default auto-close enabled no

20.10 Schema Advisors

This section describes the Schema advisors.

- [AUTO_INCREMENT Field Limit Nearly Reached](#)
- [Object Changed: Database Has Been Altered](#)
- [Object Changed: Database Has Been Created](#)
- [Object Changed: Database Has Been Dropped](#)
- [Object Changed: Function Has Been Created](#)
- [Object Changed: Function Has Been Dropped](#)
- [Object Changed: Index Has Been Created](#)
- [Object Changed: Index Has Been Dropped](#)
- [MyISAM Indexes Found with No Statistics](#)
- [Object Changes Detected](#)
- [Server-Enforced Data Integrity Checking Disabled](#)
- [Server-Enforced Data Integrity Checking Not Strict](#)
- [Object Changed: Table Has Been Altered](#)
- [Object Changed: Table Has Been Created](#)
- [Object Changed: Table Has Been Dropped](#)
- [Tables Found with No Primary or Unique Keys](#)
- [Object Changed: User Has Been Dropped](#)

AUTO_INCREMENT Field Limit Nearly Reached

Many applications need to generate unique numbers and sequences for identification purposes, such as customer IDs, bug or trouble ticket tags, membership or order numbers, and so on. 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, such as by inserting a NULL value into the AUTO_INCREMENT column, database errors are triggered 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 is not rejected by the server, but is instead changed to conform to the target column's default datatype.

**Note**

Any client can change its own session SQL mode value at any time.

For more information, see [Server SQL Modes](#).

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 performs. 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](#).

For more information, see [Server SQL Modes](#).

**Note**

Any client can change its own session SQL mode value at any time.

Default frequency 06:00:00

Default auto-close enabled no

Object Changed: Table Has Been Altered

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

Default frequency 00:10:00

Default auto-close enabled no

Object Changed: Table Has Been Created

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

Default frequency 00:10:00

Default auto-close enabled no

Object Changed: Table Has Been Dropped

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

Default frequency 00:10:00

Default auto-close enabled no

Tables Found with No Primary or Unique Keys

A primary or unique key of a relational table uniquely identifies each record in the table. Except in very unusual circumstances, every database table should have one or more columns designated as the primary key or as a unique key, and it is common practice to declare one.



Note

Tables lacking primary or unique keys can have a very negative impact on replication performance when using `binlog_format=ROW`.

Default frequency 12:00:00

Default auto-close enabled no

Object Changed: User Has Been Dropped

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

Default frequency 00:10:00

Default auto-close enabled no

20.11 Security Advisors

This section describes the Security Advisors.



Note

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

- [Account Has Old Insecure Password Hash](#)
- [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](#)
- [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 Not Being Used As Proxies](#)
- [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 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 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 plug-in, and that plug-in does not get loaded with server start, this blocks access to the database for those users.

Default frequency 06:00:00

Default auto-close enabled yes

Insecure Password Authentication Option Is Enabled



Note

This advisor does not evaluate against MySQL 8, or higher.

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.

For more information, see [LOAD DATA Statement](#).

Default frequency 00:05:00

Default auto-close enabled no

Policy-Based Password Validation Does Not Perform Dictionary Checks

If users create weak passwords 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` plug-in that can be used to test passwords and improve security. With this plug-in 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).

For more information on `validate_password` plug-in, see [The Password Validation Component](#).

Default frequency 06:00:00

Default auto-close enabled no

Policy-Based Password Validation Is Weak

If users create weak passwords, 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` plug-in that can be used to test passwords and improve security. With this plug-in 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).

For more information on `validate_password` plug-in, see [The Password Validation Component](#).

Default frequency 06:00:00

Default auto-close enabled no

Policy-Based Password Validation Not Enabled

If users create weak passwords, 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` plug-in that can be used to test passwords and improve security. With this plug-in 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).

For more information on `validate_password` plug-in, see [The Password Validation Component](#).

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 is. Starting with MySQL Server 5.6, MySQL offers a new encryption algorithm that performs authentication using SHA-256 password hashing. This is stronger encryption than that available with native authentication.

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 Not Being Used As Proxies

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.



Note

This advisor differentiates between anonymous accounts and proxy users. For more information on proxy users, see [Proxy Users](#).

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 is denied.

Default frequency 06:00:00

Default auto-close enabled no

Server Includes A Root User Account

By default, MySQL includes a root account with unlimited privileges that is typically used to administer the MySQL server. There is no reason this account must be named `root`. Accounts with this much power should not be easily discovered. Since the root account is well known in MySQL, changing its name helps prevent unauthorized users from accessing and changing the system.

Default frequency 00:05:00

Default auto-close enabled no

Symlinks Are Enabled

You can move tables and databases from the database directory to other locations and replace them with symbolic links to the new locations. You might want to do this, for example, to move a database to a file system with more free space or to increase the speed of your system by spreading your tables to different disks.

However, symlinks can compromise security. This is especially important if you run `mysqld` as root, because anyone who has write access to the server's data directory could then delete any file in the system!

Default frequency 06:00:00

Default auto-close enabled no

User Has Rights To Database That Does Not Exist

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

Default frequency 00:05:00

Default auto-close enabled no

User Has Rights To Table That Does Not Exist

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

Default frequency 00:05:00

Default auto-close enabled no

Users Can View All Databases On MySQL Server

The SHOW DATABASES privilege should be granted only to users who need to see all the databases on a MySQL Server. It is recommended that the MySQL Server be started with the `--skip-show-database` option enabled to prevent anyone from using the SHOW DATABASES statement unless they have been specifically granted the SHOW DATABASES privilege.



Note

If a user is granted any global privilege, such as CREATE TEMPORARY TABLES or LOCK TABLES, they are automatically given the ability to show databases unless the server is started with the `--skip-show-database` option enabled. DBAs should be aware of this fact, in the event that any applications make use of temporary tables.

Default frequency 00:05:00

Default auto-close enabled no

Chapter 21 GUI-Based Advisor Reference

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

21.1 Agent Health Advisor

The Agent Health Advisor monitors the monitoring agent's resource usage, communication status, backlog and memory usage.

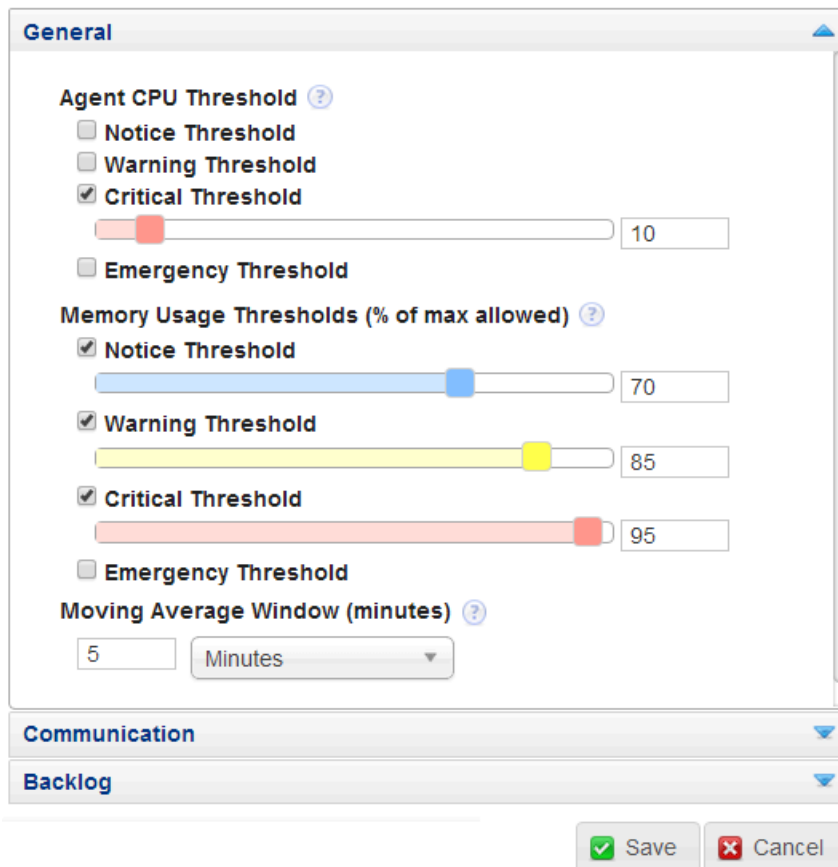
The Agent Health Advisor configuration dialog is divided into the following functional areas:

- [General](#)
- [Communication](#)
- [Backlog](#)

General

The General section defines the CPU and RAM usage thresholds. These thresholds generate events if the defined threshold value is broken by either CPU or RAM usage. Both threshold definitions use a moving average window. Although it is possible to use very small values for a moving average window, large values, larger than seconds, are recommended.

Figure 21.1 Agent Health - General



- **Agent CPU Threshold:** enables you to define thresholds for percentage CPU usage. The default value is Critical at 10% usage.
- **Memory Usage Thresholds (% of max allowed):** enables you to define thresholds for RAM usage as a percentage of the maximum heap size allocated to the monitoring agent. The default values are:
 - Notice = 70
 - Warning = 85
 - Critical = 90

Communication

The Communication section defines the thresholds for latency and HTTP errors between agent and MySQL Enterprise Service Manager.

Figure 21.2 Agent Health - General

The screenshot shows a configuration window titled 'Agent Health - General' with a 'Communication' tab selected. Under 'Agent Latency Thresholds (minutes)', the 'Warning Threshold' is set to 1 minute and the 'Critical Threshold' is set to 10 minutes. Under 'HTTP Error Thresholds (% of total requests)', the 'Notice Threshold' is set to 10%, the 'Warning Threshold' is set to 20%, and the 'Critical Threshold' is set to 30%.

- **Agent Latency Thresholds:** enables you to define thresholds for time difference between the time the data was collected and the time MySQL Enterprise Service Manager received the collected data. This can be caused by clocks that are not synchronized, network problems, and so on. The default values are:
 - Warning = 1 minute
 - Critical = 10 minutes



Important

Under certain circumstances, such as MySQL Enterprise Service Manager experiencing heavy load, events can be raised for **Agent host time out of sync relative to dashboard**. These can occur even though both MySQL Enterprise Service Manager and the monitored host are synchronized with the same time server and no time-synchronization problems exist.

The Agent Health Advisor compares the time on the MySQL Enterprise Service Manager against the time on the monitored host. If no time-synchronization issues exists, these false positive events are auto-closed.

- **HTTP Error Thresholds (% of total requests):** enables you to define thresholds for number of HTTP errors as a percentage of the total number of HTTP requests. The default values are:
 - Notice = 10
 - Warning = 20
 - Critical = 30

Backlog

If the monitoring agent is unable to communicate with the MySQL Enterprise Service Manager, it stores the collected data in memory up to a limit of 10MB, then on the filesystem, up to a limit of 10MB, giving a total limit of backlog storage of 20MB. If the limit is reached, backlog data is dropped.

Figure 21.3 Agent Health - Backlog

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

- **Backlog Memory Usage Thresholds (% of max allowed):** enables you to define a threshold for the amount of RAM used by the backlog, as a percentage of the maximum RAM allowed, 10MB. The default value is Warning = 80, which corresponds to 8MB of RAM used.

- **Backlog Disk Usage Thresholds (% of max allowed):** enables you to define a threshold for the amount of disk space used by the backlog, as a percentage of the maximum disk space allowed, 10MB. The default value is Warning = 80, which corresponds to 8MB of disk space used.

21.2 MySQL Enterprise Backup Health Advisor

This section describes the MySQL Enterprise Backup Health Advisor which checks the status of backups, and alerts according to whether they succeeded or failed.

- **Notify on succeeded or failed backups:** enables you to generate an event for the success or failure of a backup. The default values are:

- Notice = Success
- Emergency = Failure

There are no other return types.

- **Notify when incremental backups are not being used:** enables you to generate an event if the monitoring agent detects that incremental backups are not used. Select **Yes** to generate an event.
- **Notify when backup lock time is excessive:** enables you to generate an event if the backup lock time exceeds the defined thresholds. The default values are:
 - Notice = 10 seconds
 - Warning = 1 minute
- **Notify when the last successful full backup is too old:** enables you to generate an event if the last full backup is older than the defined threshold. The default value is:
 - Warning = 7 days
- **Notify when the last successful incremental backup is too old:** enables you to generate an event if the last incremental backup is older than the defined threshold. The default value is:
 - Warning = 2 days

21.3 MySQL Process Discovery Advisor

The **MySQL Process Discovery** Advisor enables you to find and, optionally, establish a connection with unmonitored MySQL instances. If you choose not to attempt a connection with the discovered instances, they are listed in the **Unmonitored MySQL Instances** list on the **MySQL Instances** dashboard.



Important

If you disable this advisor, notifications for unmonitored instances, and the associated events, are not displayed in the user interface.

Table 21.1 MySQL Process Discovery Advisor 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.

Name	Description
Admin User	<p>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 mysql schema by default.</p> <p>The SUPER privilege is required to temporarily switch off replication when creating and populating the inventory table.</p> <p>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.</p>
Admin Password	The password for the Admin User.
Auto-Create Less Privileged Users	<p>When monitoring an instance, multiple levels of user can be employed to ensure that a Process connection is not held open indefinitely.</p> <ul style="list-style-type: none"> • General User: used for general monitoring tasks that do not require SUPER level privileges, and is always connected. • Limited User: used for potentially long running statements running with SELECT only privileges. <p>If you do not have appropriate users already, they are automatically created if this option is selected. Using these lower privileged users is recommended.</p>
General User	This user handles general monitoring tasks that do not require SUPER level privileges. Lower privileged users are used instead of a SUPER user, 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	<p>Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.</p> <ul style="list-style-type: none"> • Default: uses either the server_uuid 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 host_and_datadir option can only be used when the agent is running on the same host as the MySQL instance for this connection.

Default auto-close enabled yes

21.4 Duplicate Host Identity

Tracks hosts with duplicate identities (SSH keys or Windows SIDs) by analyzing changes in a host instances' networks. The advisor triggers the condition if the same host identity changed its networks more than 4 times in 10 minutes, by default.

- **Change Rate:** number of changes per time frame.
- **Every:** time frame in which the changes are tracked.

Default auto-close enabled yes

21.5 Duplicate MySQL Server UUID

Tracks instances whose UUID is duplicated or becomes associated with multiple, different host names, or connections, over a specific time period. These changes are measured by rate, that is, by a defined number of changes over the defined time period.

- **Change Rate:** number of changes per time frame.
- **Every:** time frame in which the changes are tracked.

For example, if the **Change Rate** is set to 5, and **Every** set to 10 minutes, and the UUID of the instance changed hostname 5 times in 8 minutes, an event is generated.

Default auto-close enabled yes

21.6 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.

For more information, see [MySQL sys Schema](#).

21.7 CPU Utilization Advisor

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

CPU Usage

Figure 21.4 CPU Usage

CPU Usage

Moving Average Window ?

CPU Usage Thresholds (%) ?

Notice Threshold
 50

Warning Threshold
 75

Critical Threshold
 90

Emergency Threshold

CPU I/O Wait Thresholds (%) ?

Notice Threshold
 10

Warning Threshold
 20

Critical Threshold
 25

Emergency Threshold

- **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 21.5 CPU Outliers

The screenshot shows a configuration window for CPU Outliers. It includes the following settings:

- Enable CPU Outlier Detection**: A dropdown menu currently set to "No".
- Minimum Server Count For Outlier Detection**: A text input field containing the number "10".
- Small Group Notification**: A dropdown menu currently set to "Yes".
- Outlier Percentile**: A slider control with a numerical input field set to "90".

- **Enable CPU Outlier Detection:** Whether or not to enable the CPU outlier detection.
- **Minimum Server Count for Outlier Detection:** Minimum required sample size before outlier detection is enabled.
- **Small Group Notification:** Whether or not to generate an event if the sample size is too small to enable outlier detection.
- **Outlier Percentile:** percentage, relative to the other CPUs in the group, at which a CPU is considered an outlier.

21.8 Filesystem Free Space Advisor

Monitors and graphs the filesystem disk space usage.

Default auto-close enabled yes

General

Figure 21.6 Filesystem - General

The screenshot shows the 'General' configuration window for the Filesystem Free Space Advisor. It features a section titled 'Analyze Filesystem Types' which contains four buttons: 'Local Disk', 'Network Mount', 'RAM Disk', and 'Swap'. Each button has a small 'x' icon in its top right corner, indicating that these options are currently selected for monitoring.

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

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

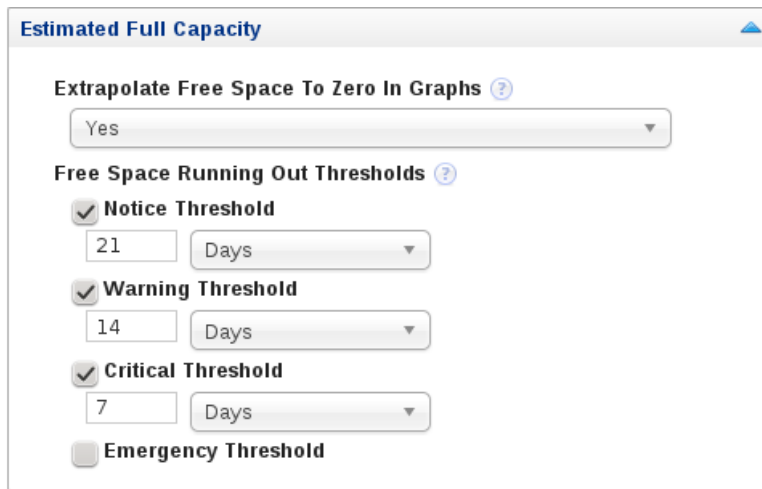
- **Swap**: enables monitoring of the system's swap file.

Select the filesystem types, as required, from the drop-down list.

To remove a filesystem type, click the **x** on the filesystem label.

Estimated Full Capacity

Figure 21.7 Filesystem - Estimated Full Capacity



Estimated Full Capacity

Extrapolate Free Space To Zero In Graphs ?
Yes

Free Space Running Out Thresholds ?

Notice Threshold
21 Days

Warning Threshold
14 Days

Critical Threshold
7 Days

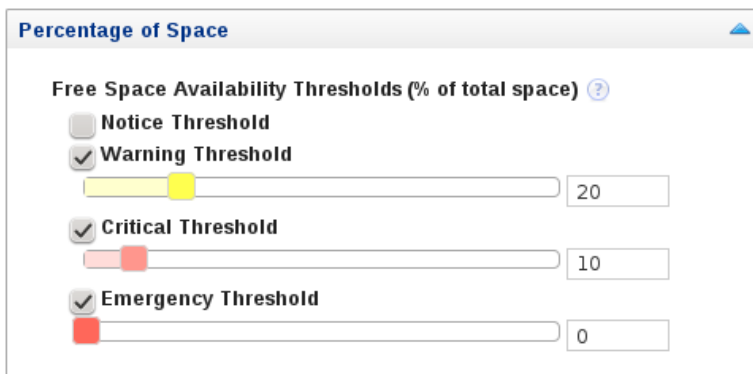
Emergency Threshold

The **Estimated Full Capacity** section monitors and graphs the time remaining to full capacity based on existing load.

- **Extrapolate Free Space to Zero in Graphs**: enables graphing of the projected time to full capacity, based on existing load.
- **Free Space Running Out Thresholds**: generate events based on when the free space is projected to run out.

Percentage of Space

Figure 21.8 Filesystem - Percentage of Space



Percentage of Space

Free Space Availability Thresholds (% of total space) ?

Notice Threshold

Warning Threshold
20

Critical Threshold
10

Emergency Threshold
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 21.9 Filesystem - Percentage Used in Time Range

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

21.9 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.

21.10 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.
- **Notification level when discovering queries with errors:** select the desired notification level for queries with errors.
- **Notification level when discovering queries with warnings:** select the desired notification level for queries with 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_statements_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 Performance Schema.

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

21.11 Group Replication Advisors

The Group Replication advisors, **Group Replication Configuration** and **Group Replication Status**, provide continuous analysis of the condition of your group replication topologies. The configuration advisor analyzes the configuration of the servers, checking for misconfiguration which could lead to unstable or insecure installations, and the status advisor continuously monitors for servers which go offline, or fall out of sync with the other members of the group.

The group replication advisors also populate the following group-level graphs:

- **Group Replication - Transactions Rows Validating - Per Member**
- **Group Replication - Transactions Checked - Per Member**
- **Group Replication - Conflicts Detected - Per Member**
- **Group Replication - Transactions In Queue - Per Member**

Group Replication Configuration Advisor

The Group Replication Configuration advisor checks the configuration of the monitored group topology and raises events if any of the group members are incorrectly or insecurely configured.

Figure 21.10 Group Replication Configuration Advisor

Select the required notification level for each configuration parameter. Event notification levels can be configured for the following:

- **Notification level for general configuration problems:** checks the following variables and alerts of potential problems:
 - `group_replication_bootstrap_group`: generates an alert if variable set to ON.
 - `group_replication_force_members`: generates an alert if variable is not empty.
 - `group_replication_allow_local_disjoint_gtids_join`: generates an alert if variable set to ON.
 - `group_replication_enforce_update_everywhere_checks`: generates an alert if variable set to OFF.

- `group_replication_single_primary_mode`: generates an alert if variable set to OFF.
- **Notification level for AUTO_INCREMENT configuration problems**: checks the `auto_increment_increment` variable, and alerts if the following conditions are not met:
 - `auto_increment_increment` is greater than, or equal to, the number of nodes.
 - `auto_increment_offset` is less than, or equal to, `auto_increment_increment`
- **Notification level for security configuration problems**: checks the following security configuration variables and alerts of potential problems:
 - `group_replication_ssl_mode`: generates an alert if variable set to OFF.

Group Replication Status Advisor

The **Group Replication Status** advisor checks the monitored topologies for errors, loss of quorum, fault tolerance, and network partitioning.

For more information, see [Group Replication](#).

Figure 21.11 Group Replication Status Advisor

Select the required notification level for each configuration parameter. Event notification levels can be configured for the following:

- **Notification level when discovering a Group Replication member with a state of ERROR**: select the required level for events if a group member's `Member_state` is `ERROR`.

For more information, see [The replication_group_members Table](#).

- **Notification level when discovering a Group Replication that is at risk of loss of quorum:** notification level if the monitored topology is found to have an even number of members.
- **Notification level when discovering a Group Replication has lost quorum:** notification level if more than half of the monitored topology's members are offline.
- **Minimum odd number of ONLINE members to be tolerant to failures:** notification level if the number of online members of the monitored topology falls beneath the defined number.
- **Notification level for network partitioning detection:** notification level if one or more members of the topology are online but unreachable by other members.

21.12 Security Advisors

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

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

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.

MySQL User Account

Analyzes the configuration of MySQL user accounts on a server and alerts when the following problems have been detected:

- **Broad Host Specifier Exclusion Regex:**

The MySQL server has user accounts with overly broad host specifiers. A MySQL account is identified by both a username and a host name, 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 host name, the account is limited to connections only from that host. If the host name contains the `%` wildcard character, the user can connect from any host that matches the wildcard character and potentially from any host at all.

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.

The advisor enables you to specify an exclusion list, using a regular expression, of users allowed to use `%` for their host entry in the `mysql.user` table. No events are generated for users matching this pattern.

For example:

```
^(user1|user2)$
```

permits only user1 or user2 to use `%` for their host entry in the `mysql.user` table. An event is generated for any other user with `%` for their host entry.

- **Global Privilege User Exclusion Regex:**

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.

This advisor enables you to define a regular expression describing the users allowed to have global privileges on the `mysql.user` table (`ALTER`, `DELETE`, `DROP`, `INSERT`, `LOCK_TABLES`, or `UPDATE`). For example:

```
^(user1@host1|user2@host2)$
```

permits only user1 on host1, or user2 on host2, to have global privileges on the `mysql.user` table. An event is generated for any other user with those privileges.

- **Strong MySQL Privilege User Exclusion Regex:**

This advisor enables you to define a regular expression describing the users allowed to have global privileges on the `mysql.user` table (`CREATE_USER`, `FILE`, `GRANT`, `PROCESS`, `LOCK_TABLES`, `SHUTDOWN`, `SUPER` or `RELOAD`). For example:

```
^(user1@host1|user2@host2)$
```

permits only user1 on host1, or user2 on host2, to have strong privileges on the `mysql.user` table. An event is generated for any other user with those privileges.

- **Database Level Privileges On All Databases User Exclusion Regex:**

Privileges such as `SELECT`, `INSERT`, and `ALTER`, allow a user to view and change data. Such operations should be limited to only those databases to which a user needs access.

This advisor enables you to define a regular expression that describes those user accounts that are allowed to have database level privileges on all databases in the `mysql.user` table (`ALTER`, `CREATE`, `DELETE`, `DROP`, `INDEX`, `INSERT`, `SELECT`, `UPDATE`). For example:

```
^(user1@host1|user2@host2)$
```

permits only user1 on host1, or user2 on host2, to have database-level privileges on all databases in the `mysql.user` table. An event is generated for any other user with those privileges.



Note

The regular expressions can be applied per instance, per server (all instances running on the server), per group (all instances in the group), or globally (all instances), depending on the asset selected.

For more information, see [Group and Host Menu](#).

21.13 NDB Cluster GUI Advisors

This section describes the NDB Cluster advisors.



Note

This section does not describe the expression-based advisors. For information in the expression-based NDB Cluster advisors, see [Section 20.4, “NDB Cluster Advisors”](#).

NDB Cluster Memory Usage Advisor

Figure 21.12 NDB Cluster Memory Usage Advisor

Parameters Schedule

Free Data Memory Thresholds (% free memory allowed) ?

Notice Threshold
25

Warning Threshold
20

Critical Threshold
10

Emergency Threshold

Free Index Memory Thresholds (% free memory allowed) ?

Notice Threshold
25

Warning Threshold
20

Critical Threshold
10

Emergency Threshold

Save Cancel

Table 21.2 NDB Cluster Memory Usage Advisor Controls

Name	Description
Free Data Memory Thresholds (% free memory allowed)	Events are generated if the percentage of free data memory in a cluster data node drops below the thresholds defined here. Database inserts start to fail as all of the memory is consumed.
Free Index Memory Thresholds (% free memory allowed)	Events are generated if the percentage of free index memory in a cluster data node drops below the thresholds defined here. Database inserts start to fail as all of the memory is consumed.

This advisor is also responsible for populating the **NDB Data Node - Data Memory** graphs.

NDB Cluster Status Advisor

Figure 21.13 NDB Cluster Status Advisor

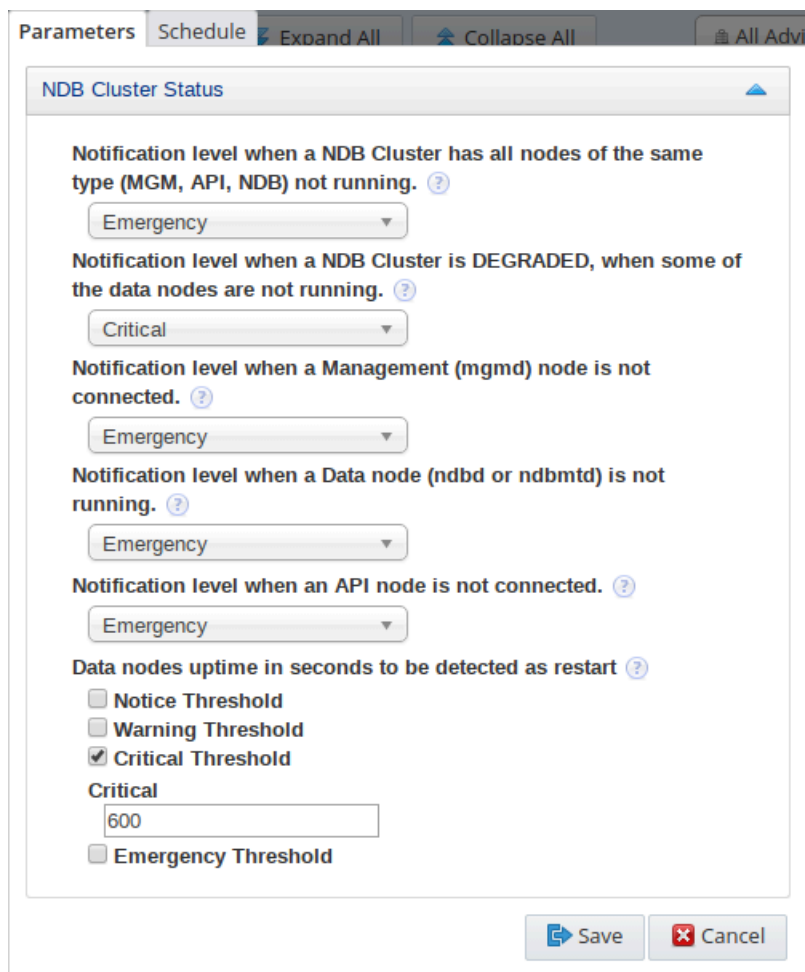


Table 21.3 NDB Cluster Status Advisor Controls

Name	Description
Notification level when a NDB Cluster has all nodes of the same type (MGM, API, NDB) not running.	Checks the node sets and generates an event if the nodes are not in CONNECTED , RESTARTNG , STARTING , STARTED , or SINGLEUSER status.
Notification level when a NDB Cluster is DEGRADED, when some of the data nodes are not running.	Checks the data-node group sets and generates events for nodes that are not in RESTARTNG , STARTING , STARTED , or SINGLEUSER status.
Notification level when a Management (mgmd) node is not connected.	Checks the node and process status and generates events if the node is not in CONNECTED status, or if the related process is not found.
Notification level when a Data node (ndbd or ndbmttd) is not running.	Checks the node and process status and generates events if either the node is not in RESTARTNG , STARTING , STARTED , or SINGLEUSER status, or if the related process is not found.

Name	Description
Notification level when an API node is not connected	Checks the node status and generates events if the node is not in CONNECTED status
Data nodes uptime in seconds to be detected as restart	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. This advisor generates events if a data node is detected that has been running for a short amount of time.

This advisor also populates the **NDB Data Nodes** graph.

Chapter 22 Event Handlers

This chapter describes 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.

22.1 Event Handlers

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

22.1.1 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.

22.1.2 Event Handlers Page

To display the Event Handlers page, select **Event Handlers** from the **Configuration** drop-down menu.

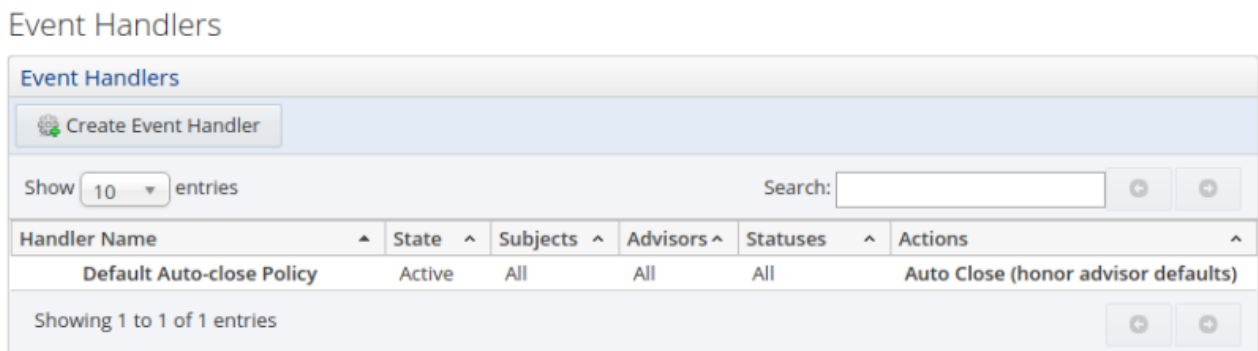
The **Event Handler** page is grouped in the following sections:

- **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.

22.1.2.1 Event Handlers List

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

Figure 22.1 Event Handlers section



Event Handlers section contains the following controls:

Table 22.1 Event Handlers Controls

Name	Description
Create Event Handler	Opens the Create Event Handler dialog. For more information, see Section 22.2, “Creating Event Handlers” .
Show * Entries	Select the maximum number of event handlers to display. This value is saved by default.
Handler Name	Lists the names of the event handlers.
State	Lists the state of the event handler. Possible states are: <ul style="list-style-type: none"> • Active: the event handler is running. • Suspended: the event handler is not running.
Groups	Lists the groups assigned to the event handler.
Assets	Lists the assets assigned to the event handler.
Advisors	Lists the Advisors assigned to the event handler.

Name	Description
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.

22.1.2.2 Email Notification Group Controls

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

Figure 22.2 Email Notification Groups section

Group Name	Recipients	Subject Line	SMS	MEM Admin
No data available in table				

Table 22.2 Email Notification Groups List 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: <ul style="list-style-type: none"> true: SMS encoding is enabled. false: SMS encoding is not enabled.
MEM Admin	Status of emails regarding MySQL Enterprise Monitor. The following values are possible:

Name	Description
	<ul style="list-style-type: none"> • true: critical MySQL Enterprise Monitor emails are sent to this notification group. • false: no email related to MySQL Enterprise Monitor are 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 Handlers page.

Figure 22.3 Create Group Dialog

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

Table 22.3 Create Group Controls

Name	Description
Group Name	Define the name of the notification groups.
Recipients	Add a comma-separated list of recipients' email addresses.
Subject Line	<p>The subject line of the notification emails. You can add some, or all, of the following variables to the subject line:</p> <ul style="list-style-type: none"> • {0}: replaced by the alarm level of the event. Critical, Warning, and so on. • {1}: replaced by the name of the advisor which generated the alert. • {2}: replaced by the name of the server which triggered the alert.
SMS	Use SMS encoding for this notification group.
MEM Admin	Identifies this group as an administration group and sends critical system messages to the defined recipients.

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 are sent.
4. In the Subject Line field, specify the subject line which are 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 are included.
7. Click Save Notification Group.

The notification group is available for use in event handlers.

22.1.2.3 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 22.4 Email Settings section

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

Table 22.4 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.

Name	Description
SMTP server	The SMTP server address. The default port is assumed unless otherwise specified. If your mail server does not use the default SMTP port, add the port number to the address using the <code>mailServerName:portNumber</code> format. For example, if your mail server, <code>mail.yourserver.com</code> , uses the port 4456, add <code>mail.yourserver.com:4456</code> to the SMTP Server field.
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.

22.1.2.4 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.

22.1.2.5 SNMP Settings

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

Figure 22.5 SNMP Settings section

The screenshot shows the 'SNMP Settings' configuration window. It includes the following elements:

- Enable SNMP Notifications
- Use SNMPv1
- Use SNMPv2
- Target 1: [Text Input] Port Number: [Text Input]
- Target 2: [Text Input] Port Number: [Text Input] (optional)
- Community String: [Text Input]
- Use the remote MySQL agent host IP address as the SNMP trap agent address for Advisor traps (optional)
- SNMP trap agent address for internally generated traps (optional): [Text Input]
- On Save send test trap (optional)
- Save SNMP Settings button

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

Table 22.5 SNMP Settings Controls

Name	Description
Enable SNMP Notifications	Activates the SNMP configuration fields.
Use SNMP v1/v2c	Choose the version of SNMP you intend to use.
Target and Port Number	IP address and Port number of the system which receive the SNMP Traps.
Community String	SNMP community string. Default value is <code>public</code> .
Use the remote MySQL agent host IP address as the SNMP trap agent address for Advisor traps (optional)	Defines the source IP address included in the trap. <ul style="list-style-type: none"> • Disabled: the trap uses the IP address of the service manager. • Enabled: the trap uses the IP address of the agent monitoring the host for which the advisor was triggered.
SNMP trap agent address for internally generated traps (optional)	Defines the source IP address included in traps generated by MySQL Enterprise Service Manager
On Save send test trap	Send a test trap message when Save is clicked. Select one, or more, of the trap types from the list. One trap is sent for each option selected.

22.1.2.6 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.

22.2 Creating Event Handlers

Event handlers enable you to create a condition which, when met, triggers notifications to concerned parties such as DBAs, System Administrators and so on.

The following condition criteria can be defined:

- **Assets and Groups:** enables you to select multiple assets or multiple groups to monitor.



Important

It is possible to define both Assets and Groups in an event handler, but is not recommended. It is recommended that you create the event handler using either Assets or Groups, not both. If you define Assets and Groups in an event handler, notifications are only sent for the defined Assets which also exist in the defined Groups.

- **Advisors:** enables you to select multiple Advisors to evaluate.
- **Event Statuses:** enables you to select multiple statuses to monitor.

To create an event handler, click **Create Event Handler** in the **Event Handlers** section on the **Event Handlers** page.

The **Create Event Handler** dialog is displayed.

Figure 22.6 Create Event Handler Dialog

Table 22.6 Create Event Handler Controls

Name	Description
Event Handler Name	Specify a name which uniquely identifies the new event handler.
Filters	
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 the selected assets only.
Assets	<p>Select the individual assets to monitor from the Assets drop-down list. If this field is left blank, all assets are included in the event handler's condition, unless one or more groups are defined. If groups are defined, and the asset field is blank, the event handler's condition includes the contents of the selected groups, only.</p> <p>The Assets drop-down list displays the Assets in their groups, if groups are defined. If no groups are defined, it lists the assets. It is not possible to select groups in the Assets field. You must expand the group to select individual assets.</p> <div data-bbox="553 1728 623 1803" style="float: left; margin-right: 10px;"> </div> <div data-bbox="760 1713 824 1743" style="font-weight: bold;"> Note </div> <div data-bbox="760 1772 1422 1902"> <p>If you select the top-level of the asset, all assets are selected. This includes network interfaces, file systems, MySQL instances, and so on. You must expand the asset's entry to select individual assets.</p> </div>

Name	Description
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	<p>Select one of the following:</p> <ul style="list-style-type: none"> • Notify on event escalation: sends a notification only if the event changes to a higher priority. From Warning to Critical, for example. No notification is sent if the event changes to a lower priority. From Critical to Warning, for example. • Notify on any status change: sends a notification if the status changes to any other status. • Always notify: sends a notification every time the defined event status(es) are broken. For example, if Critical and Emergency are defined in the Event Status field, and Always notify is selected, a notification is 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	<p>Defines whether the events are closed after the trigger status changes. The following values are possible:</p> <ul style="list-style-type: none"> • Yes: the default auto-close policy is honored and the events are closed when the conditions defined are no longer met. • No: the default auto-close policy is ignored and the events remain open in the Events page even after the conditions are no longer met.



Important

If you leave the Assets, Groups, and Advisors fields empty, the event handler can generate an extremely high volume of emails, depending on the number of assets monitored. It is recommended to create event handlers which address specific requirements and contain strictly defined criteria.



Important

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

If multiple event handlers are defined on the same advisor, their corresponding actions are combined into a single action. However, these actions are logged separately in the event handler log.

If multiple event handlers are defined on the same advisor, their corresponding actions are combined into a single action. However, these actions are logged separately in the event handler log.

22.2.1 Event Action Log

If an event handler is triggered, the action taken is displayed in the **Action Log [n]** section at the bottom of the expanded event, where [n] represents the number of actions logged for that event. For example, [Figure 22.7, “Event Action Log”](#) shows the event advice for the **MySQL Server Has Been Restarted** event with two log entries in the **Action Log**.

Figure 22.7 Event Action Log

Details:

Problem Description

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.

[Read more ▾](#)

Action Log 2

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.

Figure 22.8 Event Action Log Expanded

Action Log 43

Action time	Action Type	Successful Targets	Failed Targets	Triggering policy
Jul 10, 2018 3:02:33 PM	SMTP	Group001	None	Multiples
Jul 10, 2018 3:01:32 PM	SMTP	Group001	None	Multiples
Jul 10, 2018 2:59:32 PM	SMTP	Group001	None	Multiples

22.2.2 Suspending an Event Handler

To stop an event handler, select **Suspend Event Handler** from the event handler's drop-down menu. A prompt is displayed enabling you to enter notes on why the event handler was suspended, and confirm the suspension.



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 23 Access Control

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 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 user name, 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 View filter 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

Monitored Assets			
<input checked="" type="radio"/> System-Wide Permissions ? <input type="radio"/> Group-Specific Permissions ?			
Permission	<input checked="" type="radio"/> None	<input type="radio"/> Read-Only	<input type="radio"/> Administer
Server Group & MySQL NDB Cluster ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
MySQL Instances ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Query Analysis Aggregate Data ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Query Analysis Example and Explain Data ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
MySQL NDB Cluster API Nodes (Non-SQL) ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
MySQL NDB Cluster Data Nodes ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
MySQL NDB Cluster Management Nodes ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>



Important

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 or any of the NDB Cluster permissions to None if Server Group is set to Read-Only or higher.

23.3.1 Server Group & MySQL NDB Cluster

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.

23.3.3 MySQL NDB Cluster Permissions

This section describes the following permissions:

- **MySQL NDB Cluster API Nodes (Non-SQL):** grants access to monitored MySQL NDB Cluster API Nodes. Read-only grants permission to view details of the nodes and view events generated for the nodes. Administer permission grants the ability to edit MySQL NDB Cluster API Node details and initiate actions against the Node, and to close Events generated for the node.
- **MySQL NDB Cluster Data Nodes:** grants access to monitored MySQL NDB Cluster Data Nodes. Read-only grants permission to view details of the nodes and view events generated for the nodes. Administer grants permission to edit MySQL NDB Cluster Data Node details and initiate actions against the Node, and to close Events generated for the node.
- **MySQL NDB Cluster Management Nodes:** grants access to monitored MySQL NDB Cluster Management Nodes. Read-only grants permission to view details of the nodes and view events generated for the nodes. Administer permission grants the ability to edit MySQL NDB Cluster Management Node details and initiate actions against the Node, and to close Events generated for the node.

23.4 Monitoring Services

This section describes the **Monitoring Services** section of the Edit Role page.

Figure 23.2 Monitoring Services Permissions

Permission	<input checked="" type="radio"/> None	<input type="radio"/> Read-Only	<input type="radio"/> Administer
Agent Services access ?	<input checked="" type="radio"/>	<input type="radio"/>	(n/a)
Web Application Login ?	<input checked="" type="radio"/>	<input type="radio"/>	(n/a)
MySQL Enterprise Monitor ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Advisor Configuration ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Event Blackout ?	<input checked="" type="radio"/>	(n/a)	<input type="radio"/>
Event Handlers ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
New Group Creation ?	<input checked="" type="radio"/>	(n/a)	<input type="radio"/>
Settings ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Users and Roles ?	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

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 Handlers

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

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

- **Display Query Analyzer:** maps to **View Query Analyzer tab** in previous versions. Any user with **View Query Analyzer tab** defined, is added to the **Display Query Analyzer** role.
- **Display Query Analyzer Examples:** maps to **View actual (example) queries** in previous versions. Any user with **View actual (example) queries** defined, is added to the **Display Query Analyzer Examples** role.
- **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 is added to the following Roles:

- 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.



Note

Comma-separated strings, such as `CN=mem_manager,CN=mem_dev_manager,CN=service_manager`, are supported.

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 uses 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

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

The following scenarios are described:

- **Open:** an organization with one, or more, DBAs. All users can see, but have varying access to, all monitored assets.
- **Strict:** an organization with several DBAs and developers, and many monitored assets, grouped according to the applications and users which use them. Some users within the organization 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 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 organizations, 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 organizations, members of this role are solely responsible for the addition of users, roles and groups. While, in larger organizations, 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:

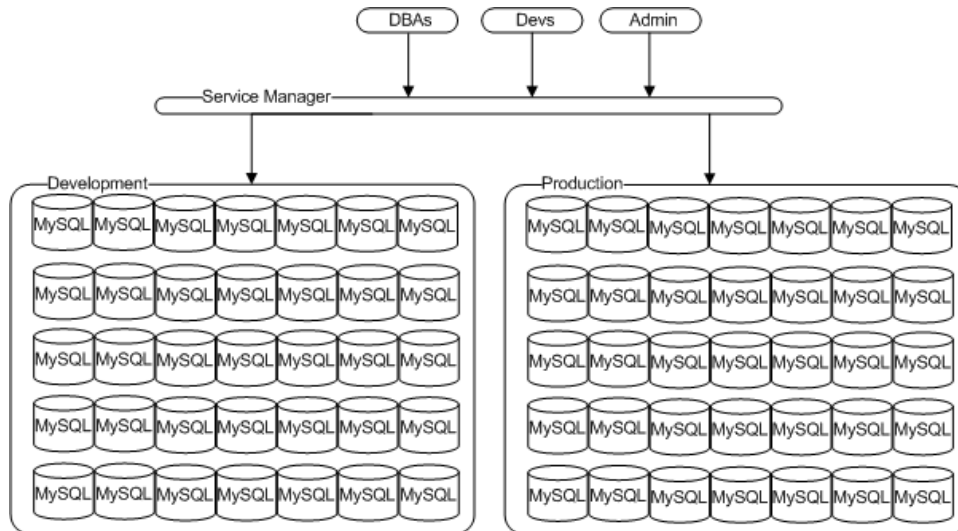
- Manager Role
 - Teamlead/Coordinator user
- DBA Role
 - DBAs

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 Strict 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, Query Analyzer 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 Query Analyzer's 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, Query Analyzer 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
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

Permission	SeniorDev
Advisor Configuration	Read-Only
Event Blackout	None
Event Handling	None
New Group Creation	None
Settings	None
Users and Roles	None

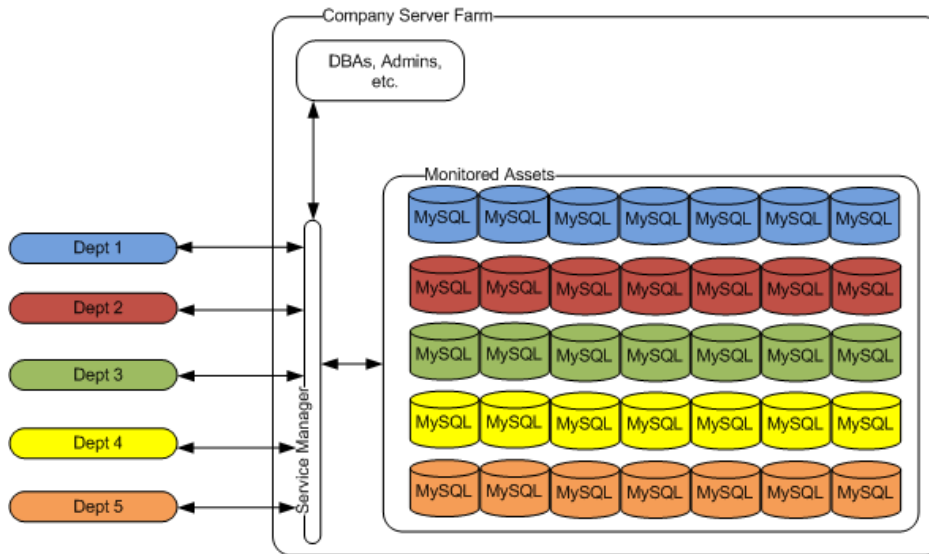
Distributed Departments

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



Part IV Using MySQL Enterprise Monitor

Table of Contents

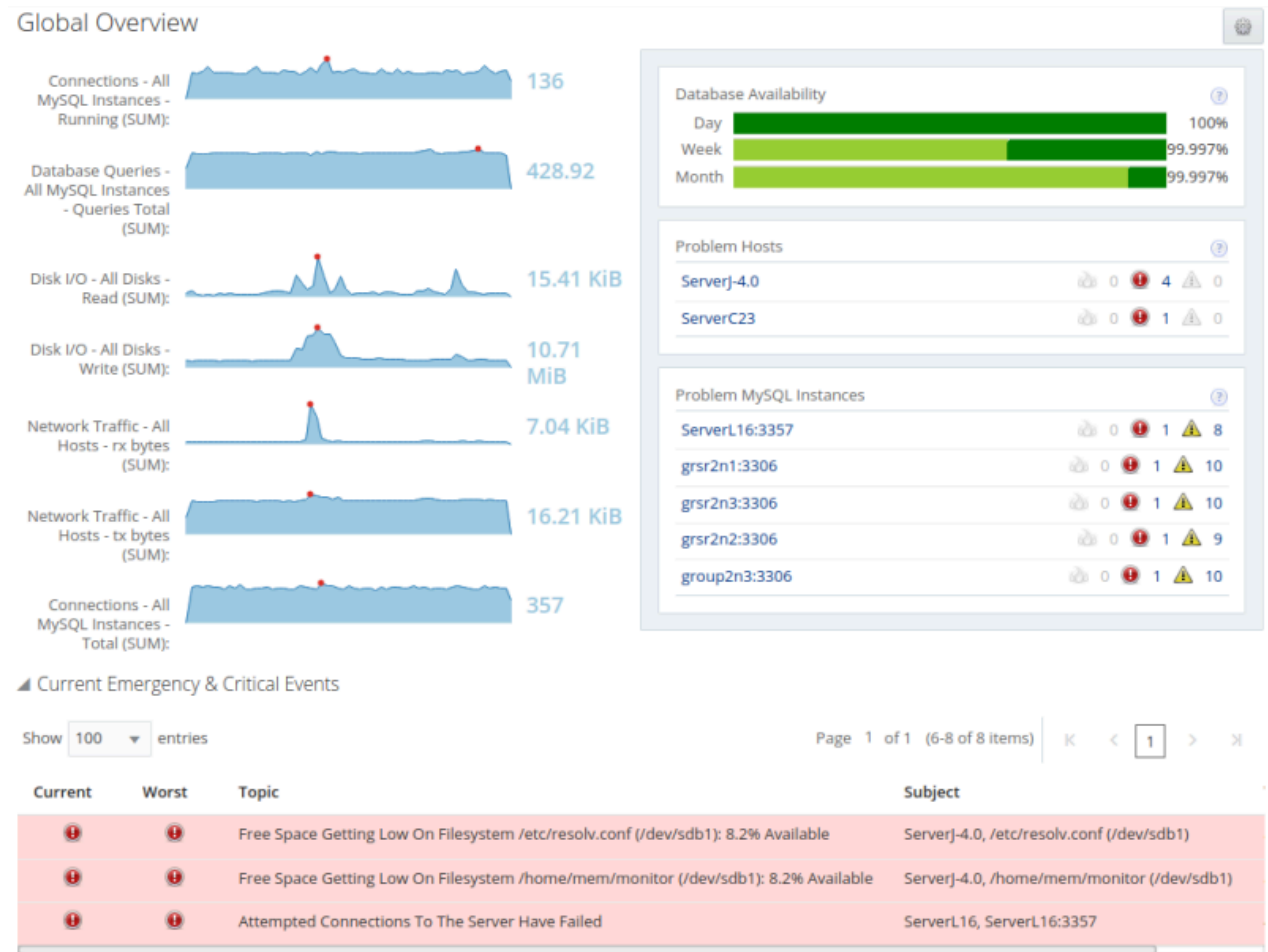
25 Overview	243
25.1 Overview Sparkline Graphs	244
25.2 Database Availability	247
25.3 General Database Statistics	248
26 Events	249
26.1 Events	249
27 Enterprise Backup Dashboard	253
27.1 Backup Overview	253
27.2 Backup Details	255
28 Replication Dashboard	259
28.1 Replication Overview	259
28.2 Replication Dashboard Tabs	260
28.3 Replication Logging	265
29 Topology	267
30 Reports and Graphs	271
30.1 All Timeseries Graphs	271
30.1.1 Graph Controls	271
30.1.2 Graph Types	272
30.1.3 Timeseries Data Aggregation	273
30.2 Database File I/O and Lock Waits	273
30.2.1 sys Schema	273
30.2.2 Database File I/O Graphs and Reports	274
30.2.3 Lock Waits Report	277
30.3 Processes Report	278
30.4 Table Statistics	278
30.5 User Statistics	280
30.6 Memory Usage Report	281
30.7 InnoDB Buffer Pool Usage	282
30.8 NDB Cluster Memory Usage	283
31 Customizing MySQL Enterprise Monitor	285
31.1 Creating Advisors and Rules	285
31.1.1 Creating Advisors	285
31.1.2 Overview of Graph Creation	286
31.1.3 Overview of Advisor Creation	286
31.1.4 Variables	288
31.1.5 Thresholds	288
31.1.6 Using Strings	289
31.1.7 Wiki Format	289
31.1.8 Creating a New Advisor: An Example	290
31.1.9 Creating a New Graph: An Example	291
31.2 Custom Data Collection	292
31.2.1 Custom.xml	292
31.2.2 Queries	293
31.2.3 Data Collection Attributes	294
31.3 Event Notification Blackout Periods	296
31.3.1 Scripting Blackouts	298

Chapter 25 Overview

The **Overview** shows a high level summary of the MySQL clusters, groups, instances, and hosts monitored by MySQL Enterprise Monitor.

The **Overview** summarizes various key statistics related to the context selection, **Database Availability**, **Problem Hosts**, **Problem MySQL Instances**, 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 25.1 Global Overview



The contents of the **Overview** depend on the selection made in the target selection menus at the top of the page. The following selections are possible:

- **Global Overview:** default context selection of **Global Summaries** and **All Targets**. the overview displays a summary of all monitored assets.
- **Group Overview:** (present only if you have created groups or are monitoring replication topologies) displays the overview of the group or topology selected from the **Global Summaries** menu. The information displayed relates to the contents of the selected group or replication topology, only.

- **MySQL InnoDB Cluster:** (present only if you are monitoring InnoDB Cluster or Group Replication topologies) displays the overview of the InnoDB Cluster or Group Replication topology selected from the **Global Summaries** menu. The information displayed relates to the contents of the selected group or replication topology, only.



Note

Group Replication topologies are included in the **InnoDB Clusters** section of the **Global Summaries** menu.

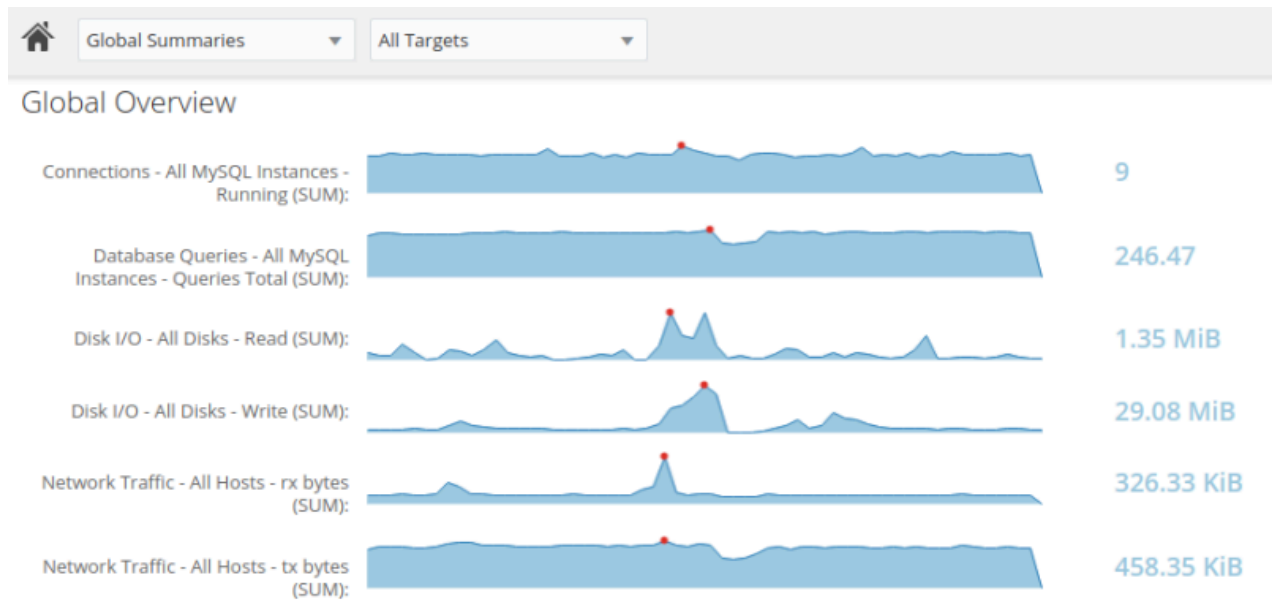
- **MySQL NDB Cluster Overview:** (present only if you are monitoring MySQL NDB Clusters) displays the overview of the MySQL NDB Cluster selected from the **Global Summaries** menu. The information displayed relates to the contents of the NDB Cluster, only.
- **MySQL Instance Overview:** displays the overview of the MySQL instance selected from the **All Targets** menu. The information displayed relates to the selected MySQL instance, only. Information on the instance's hostname, port, version, and directory paths is also displayed.
- **OS Host Overview:** displays the overview of the host selected from the **All Targets** menu. The information displayed relates to the selected MySQL instance, only. Information on the filesystems and network interfaces is also displayed.
- **NDB API Node Overview:** (present only if you are monitoring MySQL NDB Clusters) displays the overview of the NDB API node selected from the **All Targets** menu. The information displayed relates to the selected node, only.
- **NDB Data Node Overview:** (present only if you are monitoring MySQL NDB Clusters) displays the overview of the NDB Data node selected from the **All Targets** menu. The information displayed relates to the selected node, only.
- **NDB Management Node Overview:** (present only if you are monitoring MySQL NDB Clusters) displays the overview of the NDB API node selected from the **All Targets** menu. The information displayed relates to the selected node, only.
- **Agent Overview:** displays the overview of MySQL Enterprise Monitor Agent selected from the **All Targets** menu. The information displayed relates to the selected agent, only. Information on the agent's host is also displayed.

25.1 Overview Sparkline Graphs

Each group, cluster, and asset displays a different set of default sparkline graphs. The default graphs are directly relevant to the selected asset.

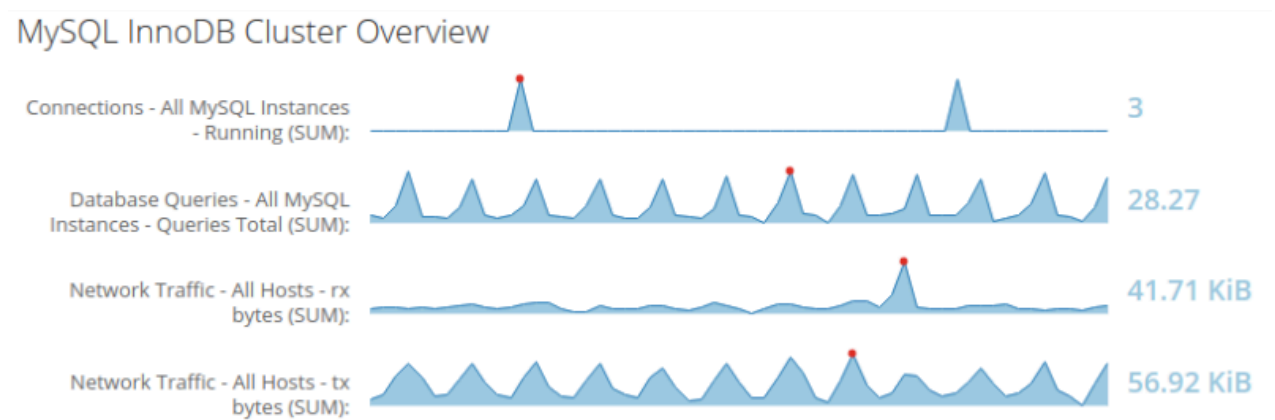
For example, the **Global Overview** displays the following by default:

Figure 25.2 Global Overview Default Sparklines



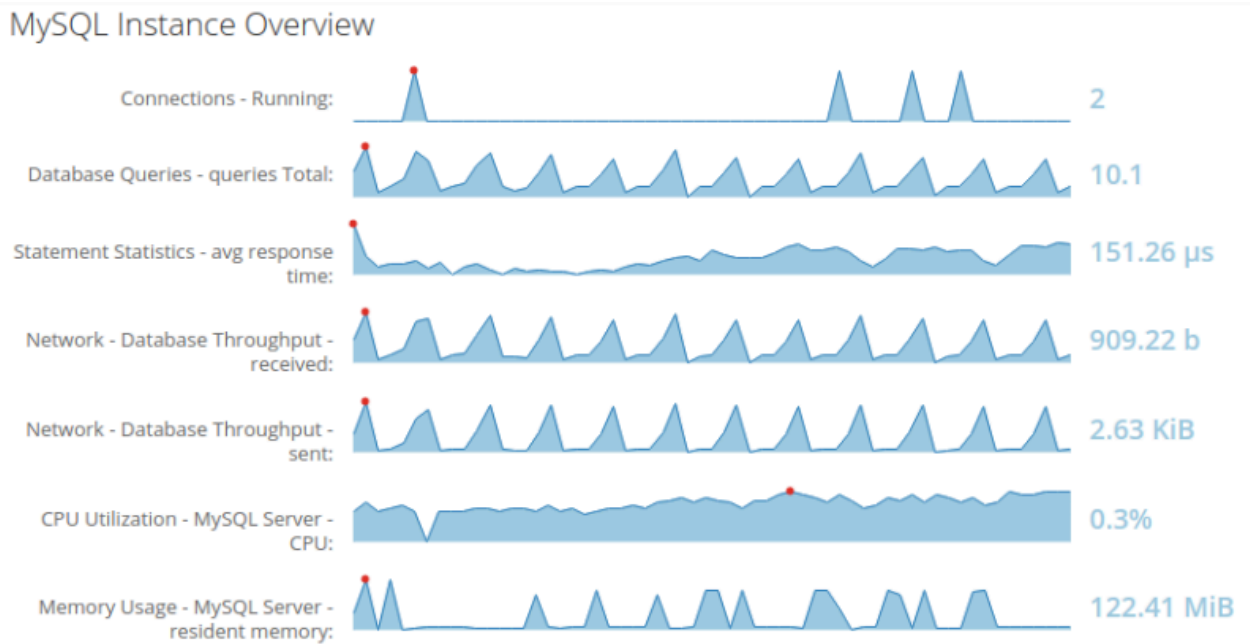
The **MySQL InnoDB Cluster Overview** displays the following graphs by default:

Figure 25.3 MySQL InnoDB Cluster Default Sparklines



If you select an individual asset, such as a MySQL Instance, the following, default graphs are displayed:

Figure 25.4 MySQL Instance Overview Default Sparklines



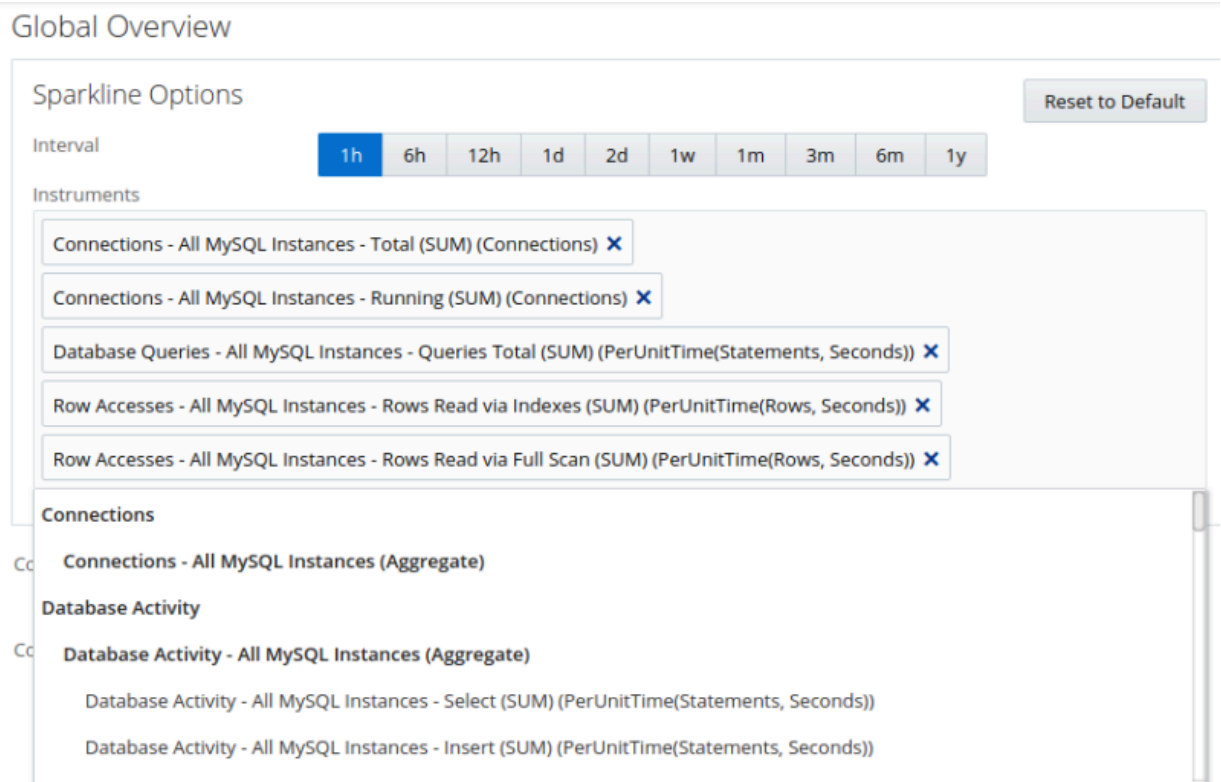
Note

The red points are the highest value, while the values in blue, at the right side of the graph, are the most recent values.

Customize the Default Graphs

The graphs are customizable. To change a graph, do the following:

1. Select the Settings button, the gear on the right side of the Overview page. The **Instruments** list is displayed:

Figure 25.5 Customize Overview Sparkline Graphs

2. Click inside the **Instruments** field and select the required graph from the drop-down list.
3. To remove a graph, click the X in the top-right corner of the graph name.
4. To close the **Instruments** list, click the gear icon again. Your selection is saved for all future sessions.
5. To reset the sparklines to the default, open the **Instruments** list and click the **Reset to Default** button.
6. To set a time interval for the graph data, select one of the interval values. The interval is saved for the selected targets.

25.2 Database Availability

The **Database Availability** statistics show an aggregate of availability statistics, generated by the **MySQL Availability Advisor**, for all MySQL Instances within the selected group, and allows you to monitor your 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

This graph is not displayed for any selection from the **All Targets** menu, except the MySQL instances.

25.3 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.



Note

These panels do not apply to every selection. For example, if a single MySQL Instance is selected from the **All Targets** menu, only the **Database Availability** graph is displayed, along with a simple listing of the instance's details.

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.

Chapter 26 Events

This chapter describes Events.

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

26.1 Events

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

Events are displayed on the **Events** view. 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 22, Event Handlers](#).

To view open events, select **Events** from the navigation bar. The target selection menus enable 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 26.1 Events Page with Filter

The screenshot shows the 'Events' page interface. At the top, there are several filter controls: 'Close Selected Events' (with a red X icon), a dropdown menu set to 'All Open Events', and a 'New' button. Below these are four dropdown menus for 'Time Range' (set to 'All Time'), 'State' (set to 'Open'), 'Current Status' (set to '(any)'), and 'Worst Status' (set to 'All Alerts'). There is also an 'Advisors' section with a text input field labeled 'Select some Advisors'. Below the filters are buttons for 'Filter', 'Discard changes', and 'Reset'. A 'Show 10 entries' control is visible, along with pagination buttons: 'First', 'Previous', '1', '2', '3', '4', '5', 'Next', 'Last'. The main content is a table with columns: 'Current', 'Worst', 'Subject', 'Topic', 'Time', and 'Actions'. The table contains 10 rows of event data, with the first five rows having red backgrounds and the last five rows having yellow backgrounds. Each row has a checkbox on the left and a red 'X' icon in the 'Actions' column. At the bottom, there is a 'Showing 1 to 10 of 94 entries' indicator and another set of pagination buttons. A 'Close Selected Events' button is also present at the bottom left.

Current	Worst	Subject	Topic	Time	Actions
!	!	Serverj-4.0, /etc/hosts (/dev/sdb1)	Free Space Getting Low On Filesystem ...	about a minute ago	X
!	!	Serverj-4.0, /etc/hostname (/dev/sdb1)	Free Space Getting Low On Filesystem ...	about a minute ago	X
!	!	Serverj-4.0, /etc/resolv.conf (/dev/sdb1)	Free Space Getting Low On Filesystem ...	about a minute ago	X
!	!	Serverj-4.0, /home/mem/monitor (/dev/...	Free Space Getting Low On Filesystem ...	about a minute ago	X
!	!	ServerL16, ServerL16:3357	Attempted Connections To The Server ...	5 minutes ago	X
!	!	Serverj-4.0, MEM 4.0 nightly mem-nigh...	Duplicate Server ID Detected - 1234567...	less than a minute ago	X
!	!	ServerL16, ServerL16:3357, MULTIPLE	SQL Statement Generates Errors or Wa...	less than a minute ago	X
!	!	Serverj-4.0, MEM 4.0 nightly mem-nigh...	SQL Statement Generates Errors or Wa...	less than a minute ago	X
!	!	ServerL16, /net/qstor01/export/backupr...	Free Space Getting Low On Filesystem ...	less than a minute ago	X
!	!	ServerL16, /build/merlin (helheim.no.or...	Free Space Getting Low On Filesystem ...	less than a minute ago	X

**Note**

You can change the column selection using the Show/Hide Columns button, adjacent to the filter drop-down list.

Event Filter

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

Table 26.1 Events Filter Controls

Name	Description
Time Range	Enables you to choose a time range. Possible values are: <ul style="list-style-type: none"> • 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.
Discard changes	Removes any changes made.
Reset to Default	Reverts any changes, and restores the saved criteria.

Events List

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







Table 26.2 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.

- **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.

Table 26.3 Event State Icons

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 22, Event Handlers](#).



Important

Not all Advisors generate events which can be auto-closed. See [Chapter 20, Expression-Based Advisor Reference](#) and [Chapter 21, GUI-Based Advisor Reference](#) for more information on the Advisors which support auto-close.

Events which support auto-close are closed by the Default Auto-close Policy after the event which triggered them is no longer in effect. For more information on

Default Auto-close Policy, see [Default Auto-close Policy](#). It is possible to override the Default Auto-close Policy by setting **Auto-Close Events** to **No** in an Event Handler

To close an individual event, click the **[X]** icon in the **Actions** column. Document the resolution using the **Notes** text area and choose the **Close Events** button. During the closing operation, you can also reconfigure the rule scheduling that triggered this event by 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 15.4, “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.

Chapter 27 Enterprise Backup Dashboard

This chapter describes the **Enterprise Backup** dashboard.

MySQL Enterprise Monitor reads the `backup_history` and `backup_progress` tables created by MySQL Enterprise Backup and presents that information on the **Enterprise Backup** dashboard. The dashboard supports Full, Incremental, TTS, and Partial backups.

27.1 Backup Overview

The following topics are described:

- [Overview Filter](#)
- [Common Backup Status Information](#)
- [Backup Overview](#)
- [Backup History](#)

Common Backup Status Information

This section describes the types of information reported on each backup. These types are used in the backup information panes.

- **Backup Events:** displays a summary of all the backup event types and lists all events for the selected groups or instance. The **Backup Events** label is a link. Clicking the link displays the **Events** page, filtered for backup events, only. To sort the events, ascending or descending, right-click in the required column and select the required sort. Each column supports sorting.
- **End Time:** date and time at which the backup completed.
- **Type:** the backup type.
- **Run Time:** duration of the backup procedure.
- **Lock Time:** total lock time during the backup process.
- **End LSN:** the InnoDB redo log sequence number at which the backup ended.
- **Binlog Position:** the point in the binlog at which the consistent snapshot was taken by the backup process.
- **Exit State:** the final state of the backup process. Values can be either `SUCCESS` or `FAILURE`.

Backup Overview

The **Backup Overview** page has two modes, Group and Instance.



Note

If no backup data is detected in the selection, an advice page is displayed.

Group Backup Overview is displayed for a group selection, while **Instance Backup Overview** is displayed for an instance selection. The titles of each frame are links to the details page of the relevant

instance. For example, clicking the **Last Successful Full Backup** title link, opens the details page for the last successful full backup. The details page opens in a tab.

- **Group Backup Overview:**

- **Last Backup:** displays the details of the last backup performed on a member of the group.
- **Last Failed Backup:** displays the details of the last backup which failed for a member of the group.
- **Last Successful Full Backup:** displays the details of the last successful full backup on a member of the group.
- **Last Successful Incremental Backup:** displays the details of the last successful incremental backup on a member of the group.
- **Instance:** (displayed on the **Group Backup Overview** only) the name of the instance, which links to the **Instance Backup Overview** page for that instance.
- **Group Backup Coverage:** displays which monitored instances have configured backups and which do not. Filtered replicas are not counted. This section also lists all instances without a configured backup. To view the list, expand the **Instances without a configured backup** link.

- **Instance Backup Overview:**

- **Last Backup:** displays the details of the last backup performed on the selected instance.
- **Last Failed Backup:** displays the details of the last failed backup on the selected instance.

If no failed backup is detected, **None found** is displayed.
- **Last Successful Full Backup:** displays the details of the last successful full backup on the selected instance.
- **Last Successful Incremental Backup:** displays the details of the last successful incremental backup on the selected instance.
- **Full Backup Run Time History:** graphs the history of the duration and lock time of all full backups run on the selected instance.
- **Incremental Backup Run Time History:** graphs the history of the duration and lock time of all incremental backups run on the selected instance.

Backup History

The **History** tab of the group view lists all backups, successful or failed, for all the instances in the group, while the instance view lists all backups performed on the selected instance. It is possible to sort the history using the column titles.

The **History** is divided into **Backups** and **Events** sections on the group-level, and the instance-level includes a **Graphs** section. The **Events** section is identical to the **Events** section on the **Current Status** tab.



Note

On first run, backup data is collected up to one month in the past.

Overview Filter

The overview filter enables you to refine the contents of the **History** page.

- **End Time:** filters the backups based on the end time of the backup, as stored in `backup_history.end_time`. Select **Interval** to use a predefined range, or select **From-To** to define a custom range.
- **Type:** the type of backup to filter on. Possible values are **Full**, **Incremental**, **TTS**, or **Partial**.
- **Exit State:** the backup result to filter on. Possible values are **Successful** or **Failed**.
- **Filter:** filter the results.
- **Reset to Default:** resets the selections and displays all results.

27.2 Backup Details

The **Backup Details** page lists the details of the selected backup. To display the **Backup Details** page, select the **History** tab and, for a group selection, click the required instance name or, for an instance selection, click the date and time entry in the **End Time** column.

The **Backup Details** page is organized in the following groups:

- [Backup Status](#)
- [Backup Configuration](#)
- [InnoDB Configuration](#)
- [Backup Command](#)
- [Progress Log](#)

Figure 27.1 Backup Details

Backup Details

Instance: order-db-replica-1, Backup ID: 14729436005405580 Back

Backup Status

Backup Type: FULL Start Time: Sep 4, 2016 12:00:00 am (3 days ago) Start LSN: 212002816 Binlog File: mysql-bin.000009 Backup Destination: /mysql-backup/2016-09-04_00-00-00	Exit State: SUCCESS End Time: Sep 4, 2016 12:00:05 am (3 days ago) End LSN: 212003530 Binlog Position: 1679
--	--

Backup Configuration

Backup Format: DIRECTORY Compression Level: 0	MySQL Data Dir: /mysql-data/ Engines: CSV:InnoDB:MyISAM:PERFORMANCE_SCHEMA
--	---

InnoDB Configuration

Data File Path: ibdata1:12M:autoextend Log Files in Group: 2 Data Home Directory:	File Format: Antelope Log File Size: 48 MiB Log Group Home Directory: /mysql-data/
--	---

Backup Command

```
/usr/bin/mysqlbackup --defaults-file=/mysql-backup/ad9cf763-710a-11e6-8546-002500a262d6.cnf --backup-dir=/mysql-backup/2016-09-04_00-00-00 --show-progress=stdout backup
```

Progress Log

Time	State	Error Code	Error Message
Sep 4, 2016 12:00:02 am	Started mysqlbackup.	0	NO_ERROR
Sep 4, 2016 12:00:04 am	mysqlbackup locking tables and copying .frm + other engines data.	0	NO_ERROR
Sep 4, 2016 12:00:05 am	mysqlbackup unlocked the tables.	0	NO_ERROR
Sep 4, 2016 12:00:05 am	mysqlbackup returns success.	0	NO_ERROR

Backup Status

- **Backup Type:** the backup type; Incremental, Full, TTS, or Partial.
- **Start Time:** date and time at which the backup started.
- **End Time:** date and time at which the backup completed.
- **Start LSN:** InnoDB log sequence number at which the backup started.
- **End LSN:** InnoDB log sequence number at which the backup ended.
- **Binlog File:** binlog file name. If no binlog is present, it is disabled, or the backup used the `--skip-binlog` option, `BINLOG-DISABLED` is displayed here.
- **Binlog Position:** the point in the binlog at which the consistent snapshot was taken by the backup process.
- **Backup Destination:** the full path to the backup files.
- **Exit State:** the final state of the backup process. Values can be either `SUCCESS` or `FAILURE`.

Backup Configuration

- **Backup Format:** the backup format.
- **Compression Level:** the level of compression used. This value ranges from 0, no compression, to 9, maximum compression.
- **MySQL Data Dir:** the path to the MySQL data directory.
- **Engines:** the storage engine file types the process is configured to backup.

InnoDB Configuration

- **Data File Path:** the InnoDB system tablespace files' path and size in backup.
- **Log Files in Group:** the number of InnoDB redo log files in the backup.
- **Data Home Directory:** the backup base directory for all InnoDB data files.
- **File Format:** the file format used, such as Barracuda.
- **Log File Size:** the total size, in bytes, of the InnoDB redo log files.
- **Log group Home Directory:** backup directory for the InnoDB redo log files.

Backup Command

Displays the full path and parameters of the backup command which created the backup.

For example:

```
meb-3-linux-64bit/bin/mysqlbackup --with-timestamp
--backup-dir=/tmp/meb-server1.30006
--backup-image=/tmp/meb-server1.30006/my.mbi
--user=MyUserName --password=xxxxxxx
--host=127.0.0.1 --port=30006
  backup-to-image --force
```

Progress Log

Displays the time-stamped progress of the backup. This is particularly useful for failed backups. The **Error Code** and **Error Message** fields provide details of the failure.

The data displayed is derived from the [mysql.backup_progress](#) table.

Chapter 28 Replication Dashboard

The Replication Dashboard displays all information related to monitored replication groups. MySQL Enterprise Monitor supports monitoring of single-source tree hierarchy, circular replication, group replication, or complex, multi-level, multi-source hierarchies.

For more information on the various types of replication supported by MySQL Server, and monitored by MySQL Enterprise Monitor, see [Replication](#) and [Group Replication](#).

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 source or replica. 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.

The **Replication** dashboard is organized by topology.



Note

Set up agents to monitor each source and replica server. Only monitored servers are displayed on this page.

The **Replication** dashboard displays all replication groups, sources, and replicas. Sources and replicas are autodiscovered and displayed based on the replication topology detected. Scans run on a five minute interval, so it can take as long as 2 polling intervals to create a complete topology.



Warning

Auto-discovery with remote monitoring is possible with MySQL 5.6 and later, only. 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 to `SHOW SLAVE STATUS` in versions preceding MySQL 5.6.

You can manage replication topologies from the **Groups** page in the same way as other groups. For more information, see [Chapter 18, Managing Groups of Instances](#). However, any replicas removed from a server group are automatically restored to that group.



Important

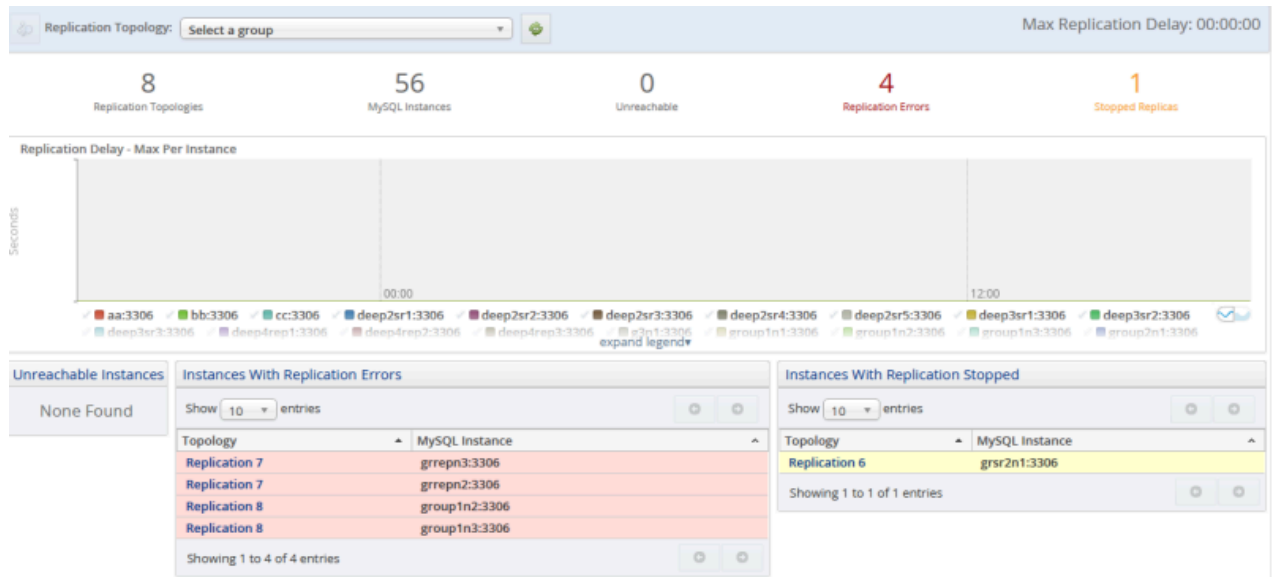
For information on the kinds of problems you might find while monitoring replication, and how to solve them, see [Troubleshooting Replication](#) and [Improving Replication Performance](#).

28.1 Replication Overview

The Replication Overview page displays the current status of all monitored replication topologies.

The overview is displayed if you select **Replication** on the **Dashboards** menu for the first time in a session. To open the overview while viewing a replication topology, select the **Show Replication Overview** button adjacent to the **Replication Topology** selection menu.

Figure 28.1 Replication Overview



The overview displays the following:

- **Topologies at a glance:** displays the numbers of
 - **Replication Topologies:** the total number of monitored replication topologies.
 - **MySQL Instances:** the total number of instances in the replication topologies.
 - **Unreachable:** the number of instances which were monitored but are not currently responding.
 - **Replication Errors:** the total number of replication errors.
 - **Stopped Replicas:** the total number of replicas which are no longer running.
- **Replication Delay - Max Per Instance:** graphs the replication delay for all monitored nodes.
- **Unreachable Instances:** lists the instances which were monitored but are not currently responding.
- **Instances With Replication Errors:** lists the names of all the instances reporting replication errors, and the groups to which they belong. To open the group, click on the group name.
- **Instances With Replication Stopped:** lists the names of all the instances no longer replicating, and the groups to which they belong. To open the group, click on the group name.

Refresh intervals are disabled when the Replication Overview is displayed.

28.2 Replication Dashboard Tabs

This section describes the tabs of the Replication Dashboard.



Important

The **Max Replication Delay** field displays the total amount of replication delay which exists on the monitored topologies.

To display a replication group, choose a group from the **Replication Topology** drop-down list.



Important

Clicking the refresh icon, adjacent to the **Replication Topology** drop-down list, forces a re-discovery of all replication groups.

Status

The **Status** tab displays information on the status of the members of the replication topologies. Each instance expands to display detailed information in a drill-down.



Note

Data displayed in red indicates a problem with your configuration, while green indicates preferred settings.

If the monitored topology is a group topology, a status message is displayed above the **Status** table. The status message summarizes the current state of the group, whether all members are online, fault tolerance, and whether the group has quorum. The status message is prefixed with the Group Replication ID, which takes the form GR-nnnnnnnn-nnnn, where nnnnnnnn-nnnn is the UUID prefix of the group.

For more information on Group Replication, see [Group Replication](#).

Table 28.1 Group Status Columns

Name	Description
Instance	(All topologies) Name of the monitored instance.
Member State	(Group Replication topologies only) Displays whether the member is ONLINE , ERROR , RECOVERING , OFFLINE , or UNREACHABLE . N/A is displayed if the member is not using the Group Replication plugin.
Fetch State	(All topologies) Displays the current status of the fetch operations. If the replica successfully retrieved data from the source, the icon is green. If the replication has not stopped because of an error, the icon is yellow. If the state is unknown, either because the instance is currently unmonitored, or no replication is taking place, the icon is red and the errors are displayed in a tooltip.
Apply State	(All topologies) Displays the current status of the apply operations. If the changes are successfully applied to the replica, the icon is green. If the apply operation was stopped, but no error occurred, the icon is yellow. If the apply operation stops because of an error, the icon is red and the error displayed in a tooltip.
Time Behind	(Asynchronous topologies only) Amount of time the replica is behind the source. N/A is displayed for sources. If there is a problem with the Fetch or Apply state, Unknown is displayed.
Read Only	(All topologies) Displays one of the following values: <ul style="list-style-type: none"> • ON: instance is configured as read_only • SUPER: instance is configured as super_read_only • OFF: instance is a source.
GTID Enabled	(Asynchronous topologies only) Displays if the server is using GTIDs.

Name	Description
Binary Log Format	(Asynchronous topologies only) Displays the format used by the binary log.
Node Type	(All topologies) Displays the type of node. Source, Replica, Primary Group Member, Secondary Group Member, and so on. If the replication type is multi-source, the node can be Replica/Source.
Flow Control	(Group Replication only) The mode used for flow control. Possible values are QUOTA or DISABLED. N/A is displayed for instances which are not using the Group Replication plugin.
View ID	(Group Replication only) Unique identifier of the view.
Group Auto Increment	(Group Replication only) The auto increment defined for the group's members.
Channels	(Asynchronous topologies only) Displays the number of channels used by this instance.
Version	(All topologies) Displays the version of the MySQL instance.

The instance drill downs contain different information depending on if the instance is a source or a replica, and also depends on the type of replication in use. Each replica also contains a tab listing all the instances from which it is receiving updates.

- Source drill down contents:
 - **Node Status:** graphs the source load in running connections, queries per second, and the average response time. The server IDs are also provided.
 - **Group Replication Status:** (Group Replication only) graphs the transaction status in transactions queued, validating, checked, and in conflict.
 - **Local Binary Log Status:** displays the details of the local binary log, such as name, position, and GTID set.
 - **Binary Log Statistics:** graphs the average read/writes per second, log file latency and total space usage.



Note

This frame requires sys schema to display all graphs. If sys schema is not installed on the node, only the Total Space Usage graph is displayed.

- **Most Appropriate Candidates for Source Promotion (best candidate on top):** lists the replicas which are eligible for promotion to source.

For a replica to be eligible for promotion to source, it must have the same GTID mode as the source. If GTID mode and binary logging are disabled, the replica cannot be promoted. Such replicas are listed in red with a message explaining why they are not eligible for promotion.

Ordering of candidates is based on a scoring system. Replicas with the highest value of `RECEIVED_TRANSACTION_SET` in the `performance_schema.replication_connection_status` table are promoted.

- **Semi-Synchronous Replication:** this frame is present for both asynchronous and semi-synchronous types, but is only populated if semi-synchronous replication is in use. If semi-synchronous replication is running, this section contains graphs showing the Sessions Waiting, Waits, and transaction status.

- **Replication Configuration:** displays all information related to the replication configuration:
 - **General**
 - **Group Replication Recovery**
 - **Networking**
 - **Relay Logging**
 - **Group Replication General**
 - **Group Replication Security**
 - **Group Replication Performance**
 - **Crash Safety**
 - **Appliers**
 - **Multi-Threaded Appliers**
- Replica drill-down contents:
 - **Node Status:** graphs the replica load in running connections, queries per second, and the average response time. The server IDs are also provided.
 - **Source Replication Status:** displays a tab for every source replicating to the selected replica. Each tab displays the following:
 - **Fetch Status:** displays graphs and information on the status of the data retrieval from the source.
 - **Filter / Delay Status:** displays the defined and remaining delays and any filtered databases or specific tables.
 - **Apply Status:** displays the current apply status, the last executed GTID set, and so on. The status by worker is also displayed.



Note

The **Info** column of the **Status by Worker** section requires the user have the **Query Analysis Example and Explain Data** permission set to at least Read-Only. If this permission is not granted, the **Info** column cannot be populated.

- **Source Configuration:** displays the instance name, port, and username of the source.
- **Channel Configuration:** displays the channel configuration information.
- **Secure Connection Configuration:** displays all SSL-related information for the monitored instance.
- **Replication Configuration:** displays all information related to the configuration of the replication process.

It is also possible to edit and delete connections from this tab. Clicking the drop-down list next to the instance opens the instance context menu. For more information on this menu, see [Group and Instance Context Menu](#).

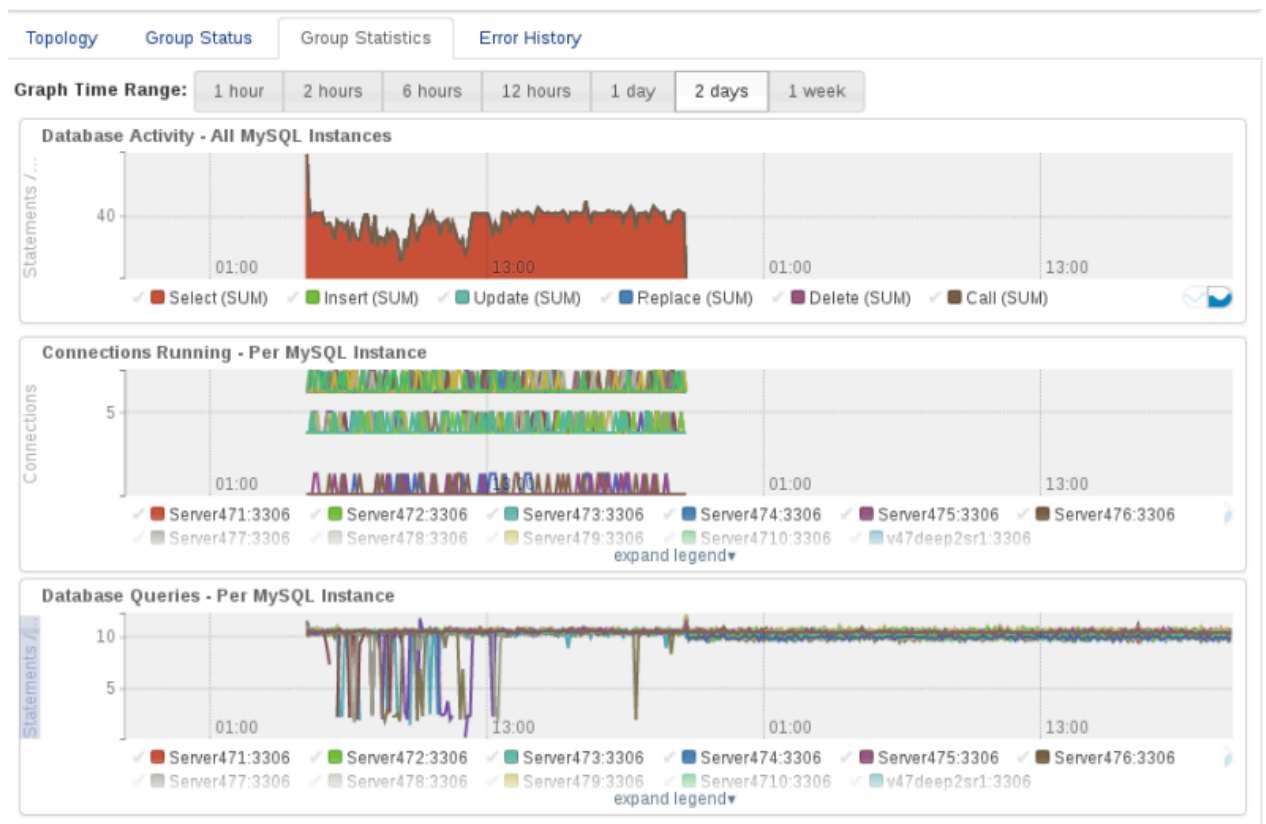
Statistics

The **Statistics** tab graphs the activity of all members of the topology. You can change the data displayed using the **Graph Time Range** selections. If a range is selected for which there is no data, no graph is displayed.

The following graphs are available:

- **Database Activity - All MySQL Instances:** displays the total number of `SELECT`, `INSERT`, `UPDATE`, `REPLACE`, `DELETE`, and `CALL` statements performed on all members of the selected group.
- **Connections Running - Per MySQL Instance:** displays the running connections for each member of the group.
- **Database Queries - Per MySQL Instance:** displays the number of queries performed against each member of the group.

Figure 28.2 Replication Statistics tab



If the selected topology is a member of a group replication topology, the following graphs are also displayed:

- **Group Replication - Transactions Rows Validating - Per Member**
- **Group Replication - Transactions Checked - Per Member**
- **Group Replication - Conflicts Detected - Per Member**
- **Group Replication - Transactions In Queue - Per Member**

Error History

The **Error History** tab displays all errors logged for the members of the selected group. The time of the error, type of error and error message are displayed.

28.3 Replication Logging

Discovery events are logged to the `mysql-monitor` log files. This log can be a useful tool for debugging the replication topology discovery process.

To enable replication logging, add the following `Logger` element to the `log4j2.xml` file:

```
<Logger name="com.mysql.replication" level="debug"/>
```

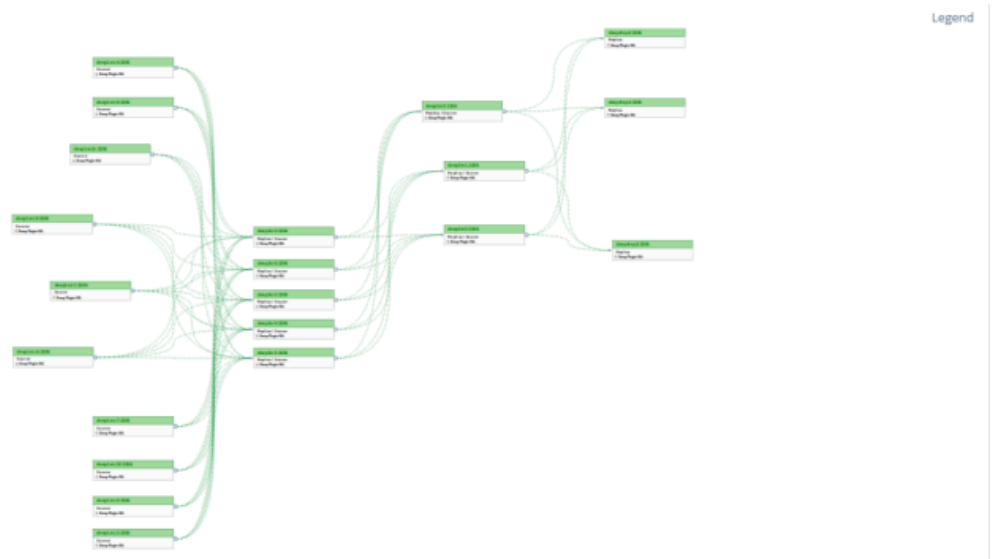
For additional information about log4j, read the log4j documentation at <http://logging.apache.org/log4j/>.

Chapter 29 Topology

This chapter describes the Topology view. Topology is only available on the navigation bar if a replication topology, InnoDB Cluster, or NDB Cluster is selected.

The **Topology** tab graphically displays the replication topology of the selected group.

Figure 29.1 Replication Topology Graph



The Topology view contains the following:

- Topology graph: graphically displays the replication setup, enabling you to reposition the nodes for better visibility, and zoom in and out using your mouse wheel.
- Tooltips: For information on an individual node or channel, hover the mouse cursor over the node or channel. A tooltip is displayed containing information about the node or channel. The node tooltip displays the ID, Name, and Status of the node, while the channel tooltip displays the name of the channel.
- **Legend** : displays the meaning of each of the graphical elements.
 - **Nodes:**
 - **Node OK**
 - **Replication Issues**
 - **Node/Replication Down**
 - **Node Unmonitored**
 - **Links:**
 - **Async Fetch OK**
 - **Async Fetch Down**

- **Semi-Sync Fetch OK**
- **Semi-Sync Fetch Down**
- **Group Replication Recovery**
- **Link Status Unknown**
- **Group replication Status:**
 - **Online**
 - **Group Replication Recovering**
 - **Offline/Error**
 - **Unknown/Plugin Not Available**

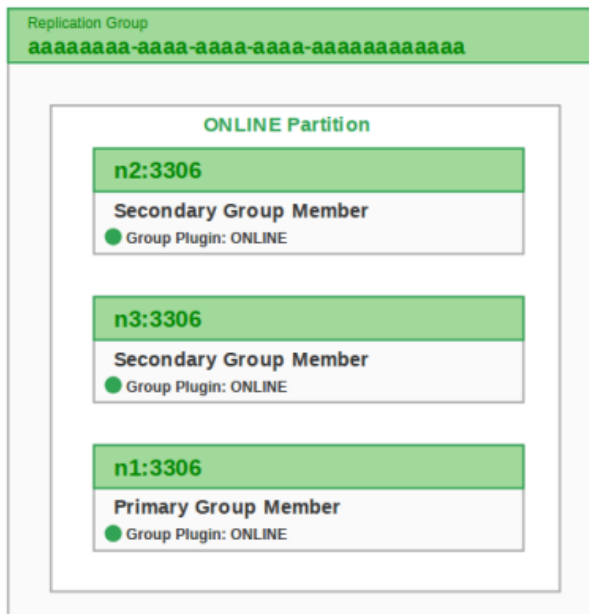


Note

The topology graph displays semi-synchronous links only if both source and replica use the semi-sync plugins.

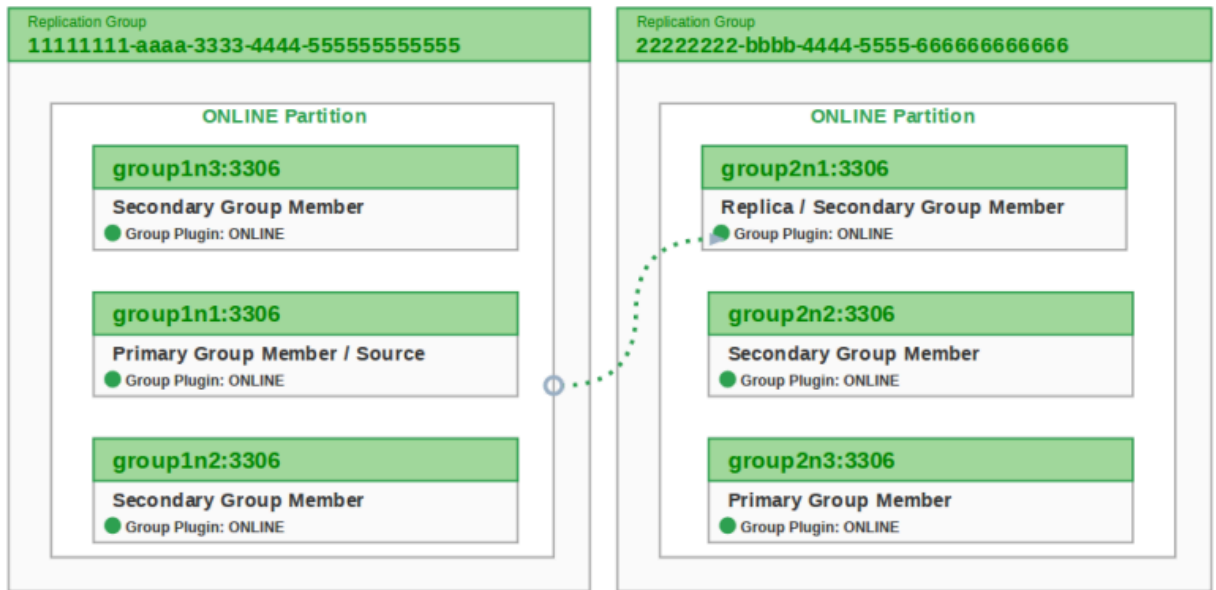
Group replication topologies are displayed differently. For example, the following is a three-node group:

Figure 29.2 Group Replication Single Group Topology



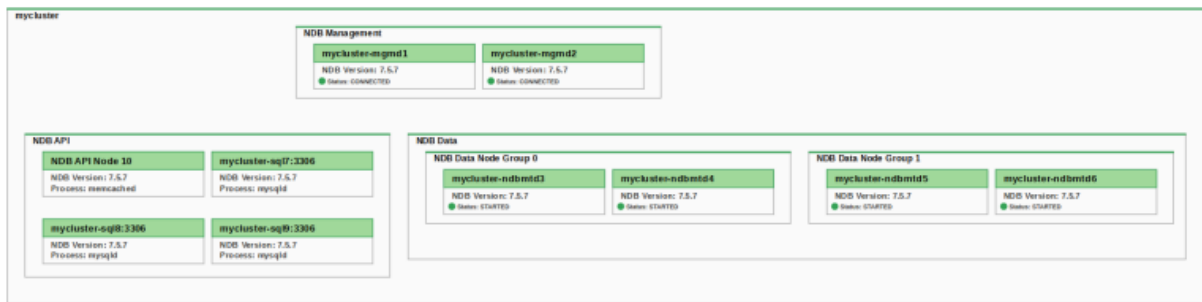
While the following shows normal communication between two groups:

Figure 29.3 Group Replication Multiple Group Topology



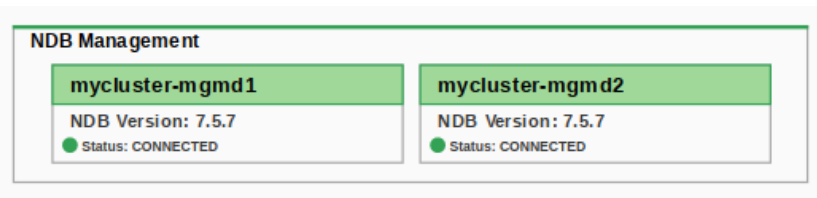
NDB Cluster topologies contain all the cluster components, Management node, API node, and data node, and uses color to display their states. Each layer is identified by host or process information. Further information can be seen by hovering the cursor over any of the elements.

Figure 29.4 NDB Cluster Topology



NDB Management Node:

Figure 29.5 NDB Cluster Topology - Management Node



NDB API Node:

Figure 29.6 NDB Cluster Topology - API Node

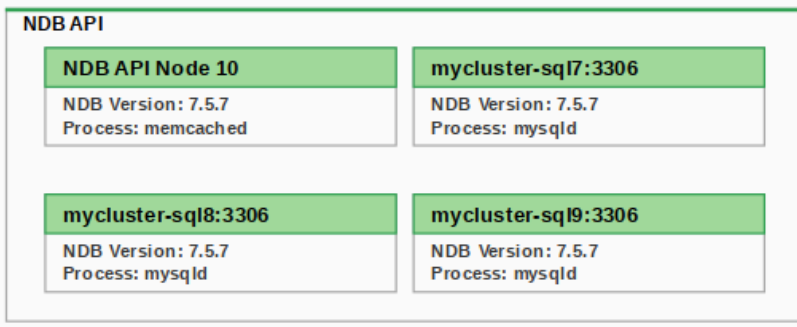
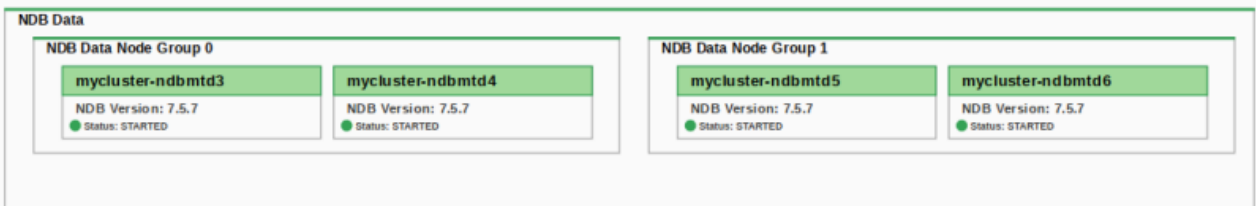


Figure 29.7 NDB Cluster Topology - Data Node



Chapter 30 Reports and Graphs

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

30.1 All Timeseries Graphs

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

30.1.1 Graph Controls

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

View Filter

The contents of the **Timeseries Graphs** page depend on the selections made in the View filter. The View filter is a pair of drop-down menus on the top-left side of the page. They enable you to drill down into the assets to which you have access.

For example, if you select a group in the **Global Summaries** drop-down list, and a MySQL instance in the **All Targets** drop-down list, the graphs displayed are the instance-specific graphs, only. If you select a host in the **All Targets** drop-down list, only host-specific graphs are displayed.



Important

The View filter displays only those assets to which you have access.

Graph Filter

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

Table 30.1 Timeseries Graph Filter

Name	Description
Graph Name	Opens a drop-down menu listing the available search types: <ul style="list-style-type: none">• Contains• Doesn't Contain• Regex• Negative Regex
Value	Free text field for the search term or regular expression.
Time Range	Drop-down lists containing the time periods to apply to the graphs. The possible values are: <ul style="list-style-type: none">• Interval: select the duration for the overview data. If you select 1 hour, the data collected in the last hour is displayed.• From/To: select a date and time range for the overview data.
Filter	Apply the defined filter.
Save as Default	Sets the selected group and time range as the default.

Name	Description
	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. Several icons are displayed in the top-right corner of the selection. An x to close the selection, a database icon, and a magnifying glass. Click the database icon to open the Query Analyzer's **Browse Queries** page. Click the magnifying glass to zoom in on the selected range.

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 32.3, "Query Analyzer User Interface"](#).

30.1.2 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 30.7, “InnoDB Buffer Pool Usage”](#) for an example.

30.1.3 Timeseries Data Aggregation

This section describes the graph data aggregation process.

When you upgrade to MySQL Enterprise Monitor 3.4, all existing timeseries data is aggregated when the upgraded MySQL Enterprise Service Manager first starts. A progress bar is displayed on the user interface, adjacent to the system status bar, indicating the completeness of the aggregation process.

Figure 30.1 Optimizing Graph Data Status



Note

While the timeseries data is being aggregated for the first time, no performance enhancements are evident in the graph response times. Performance improvement is evident only after the process completes and the timeseries data is retrieved from the new, aggregated data.

For new installations of MySQL Enterprise Monitor 3.4, data aggregation begins when MySQL Enterprise Service Manager first starts.

Timeseries data aggregation is an ongoing process. As the data is collected from the monitored instances and hosts, it is aggregated in real time, hourly, and daily.

30.2 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.



Note

This graph is displayed only if a MySQL instance is selected in the All Targets drop-down menu.

30.2.1 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](#).

**Important**

If you have installed an older version of `sys` schema on your monitored instances, it is recommended to upgrade to the latest version. The upgrade must be performed from the command line. It is not currently possible to upgrade `sys` schema from MySQL Enterprise Service Manager.

30.2.2 Database File I/O Graphs and Reports

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

**Note**

This report requires `sys` schema be installed on the selected MySQL instance. If `sys` is not installed, and the selected instance is compatible, you are prompted to install it.

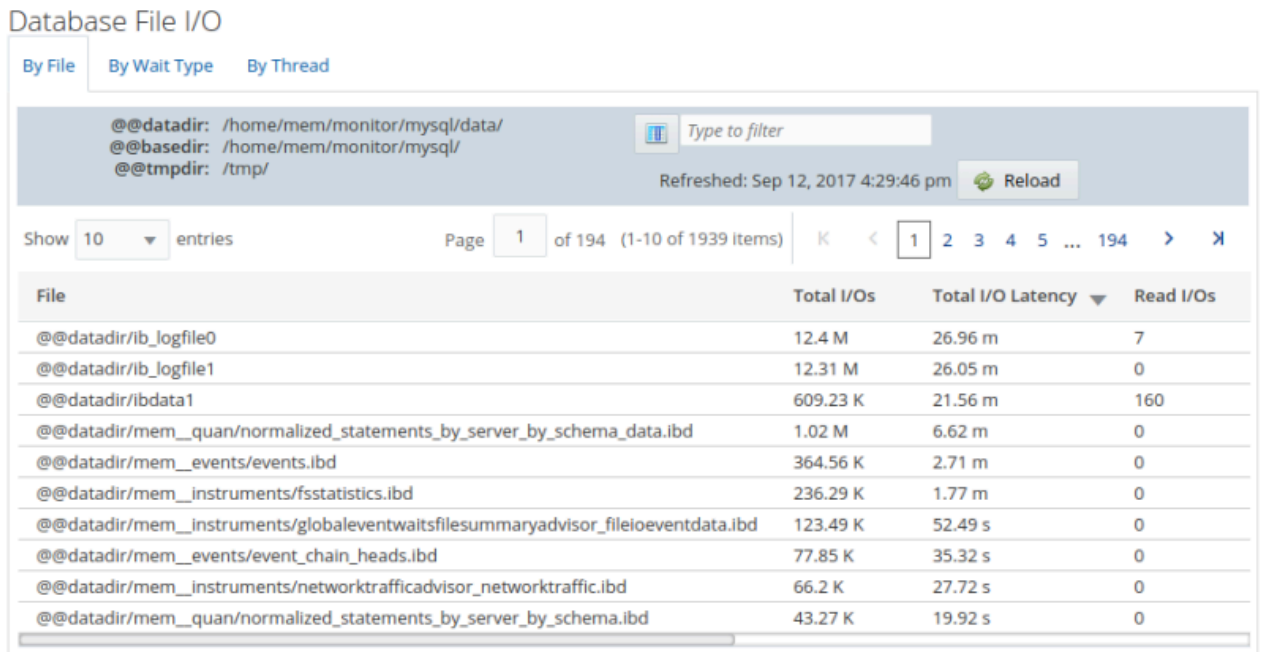
Each tab contains the following common elements:

- **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.

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.

For more information on the data retrieved in this report, see [The io_global_by_file_by_latency and x \\$io_global_by_file_by_latency Views](#).

Figure 30.2 Database File I/O By File

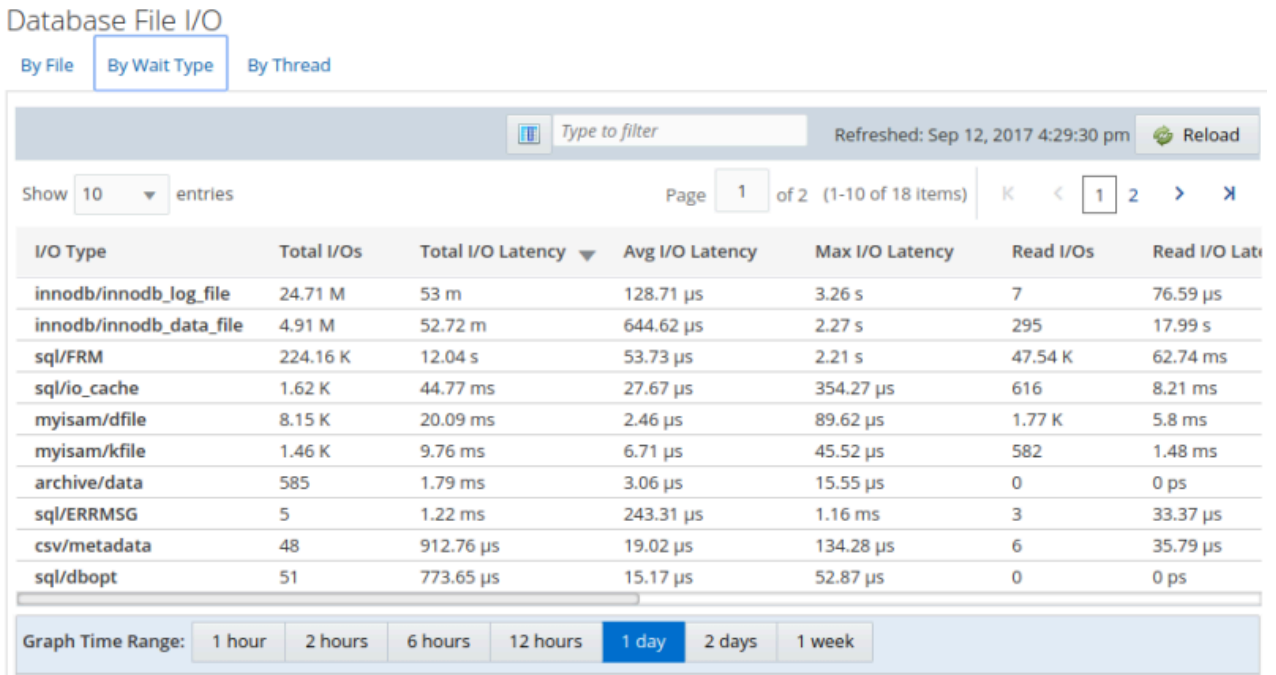
I/O By Wait Type

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.

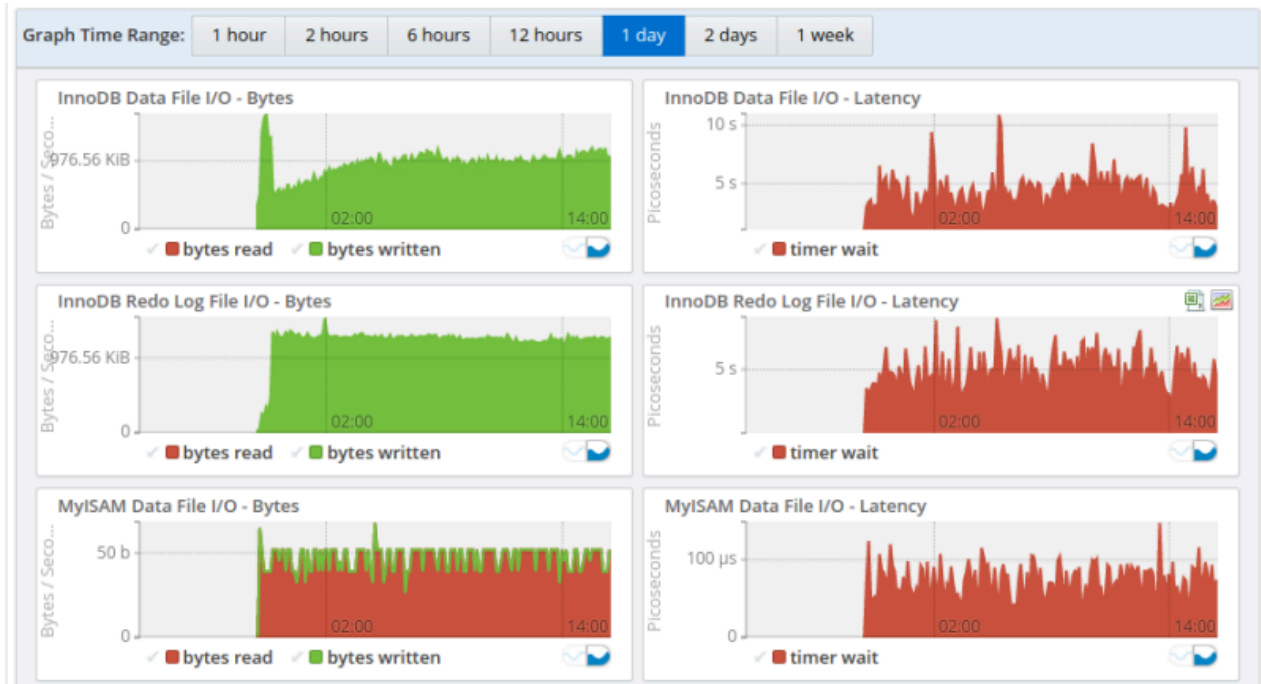
For more information on the data retrieved in this report, see [The io_global_by_wait_by_latency and x \\$io_global_by_wait_by_latency Views](#).

Figure 30.3 Database File I/O By Wait Type Report



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

Figure 30.4 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.

For more information on the data retrieved in this report, see [The io_by_thread_by_latency and x \\$io_by_thread_by_latency Views](#).

Figure 30.5 Database File I/O By Thread

Database File I/O

By File By Wait Type **By Thread**

Type to filter Refreshed: Sep 12, 2017 4:33:09 pm Reload

Show 10 entries Page 1 of 4 (1-10 of 36 items) K < 1 2 3 4 > X

Connection ID	Account	Total I/Os	Total I/O Latency	Avg I/O Latency	Max I/O Latency
101925	root@localhost	2	9.59 µs	4.8 µs	5.23 µs
101924	root@localhost	1	14.13 µs	14.13 µs	14.13 µs
101923	root@localhost	4	35.98 µs	8.99 µs	12.7 µs
101922	root@localhost	4	21.54 µs	5.39 µs	8.17 µs
101921	root@localhost	6	37.86 µs	6.31 µs	8.8 µs
101920	root@localhost	1	9.16 µs	9.16 µs	9.16 µs
101883	root@localhost	1.31 K	12.43 ms	9.51 µs	48.02 µs
101882	root@localhost	447	4.64 ms	10.39 µs	34.89 µs
101881	root@localhost	821	7.44 ms	9.07 µs	38.73 µs
101880	root@localhost	744	7.57 ms	10.18 µs	39.92 µs

30.2.3 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`.

For more information on the data retrieved in this report, see [The innodb_lock_waits and x \\$innodb_lock_waits Views](#).

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.

The P_S instrument `wait/lock/metadata/sql/mdl` must be enabled to use this report. It is disabled by default. To enable this instrument, add the following to the `my.cnf` file of the monitored server:

```
[mysqld]
performance-schema-instrument='wait/lock/metadata/sql/mdl=ON'
```

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

For more information on the data retrieved in this report, see [The schema_table_lock_waits and x \\$schema_table_lock_waits Views](#).

30.3 Processes Report

The Processes report lists the processes currently running on the selected MySQL instance, with their Thread ID, User, and so on.



Note

This report requires sys schema be installed on the selected MySQL instance. If sys is not installed, and the selected instance is compatible, you are prompted to install it.



Note

This graph is displayed only if a MySQL instance is selected in the All Targets drop-down menu.

For more information on the data retrieved by this report, see [The processlist and x\\$processlist Views](#).

Figure 30.6 Processes Report

Processes

[MySQL Processes](#)

Type to filter Refreshed: Sep 12, 2017 4:28:13 pm Reload

Show 10 entries Page 1 of 6 (1-10 of 51 items) K < 1 2 3 4 5 6 > X

Thread ID	Connection ID	User	Database	Command	State	Time	Statement L
3443	3418	root@localhost	mysql	Query	Sending data		2.09 ms
26		sql/signal_handler					
101498	101473	root@localhost	mysql	Sleep		39s	
101762	101737	root@localhost	mem	Sleep		7s	
101764	101739	root@localhost	mem	Sleep		5s	
27	2	sql/compress_gtid_table		Daemon	Suspending	17h 55m 7s	
101765	101740	root@localhost	mem	Sleep		5s	
1		sql/main				17h 55m 7s	
101815	101790	root@localhost	mem	Sleep		5s	
101816	101791	root@localhost	mem	Sleep		5s	

30.4 Table Statistics

The **Table Statistics** report is based on the table statistics sys schema view, [schema_table_statistics](#). For more information, see [The schema_table_statistics and x\\$schema_table_statistics Views](#).



Note

This graph is displayed only if a MySQL instance is selected in the All Targets drop-down menu.

Table View

The table view lists the statistics in a standard table format. The data can be sorted by column.

Figure 30.7 Table Statistics Report - Table View

Table Statistics

Table View Treemap View

Refreshed: Oct 19, 2023 8:17:17 pm Filter

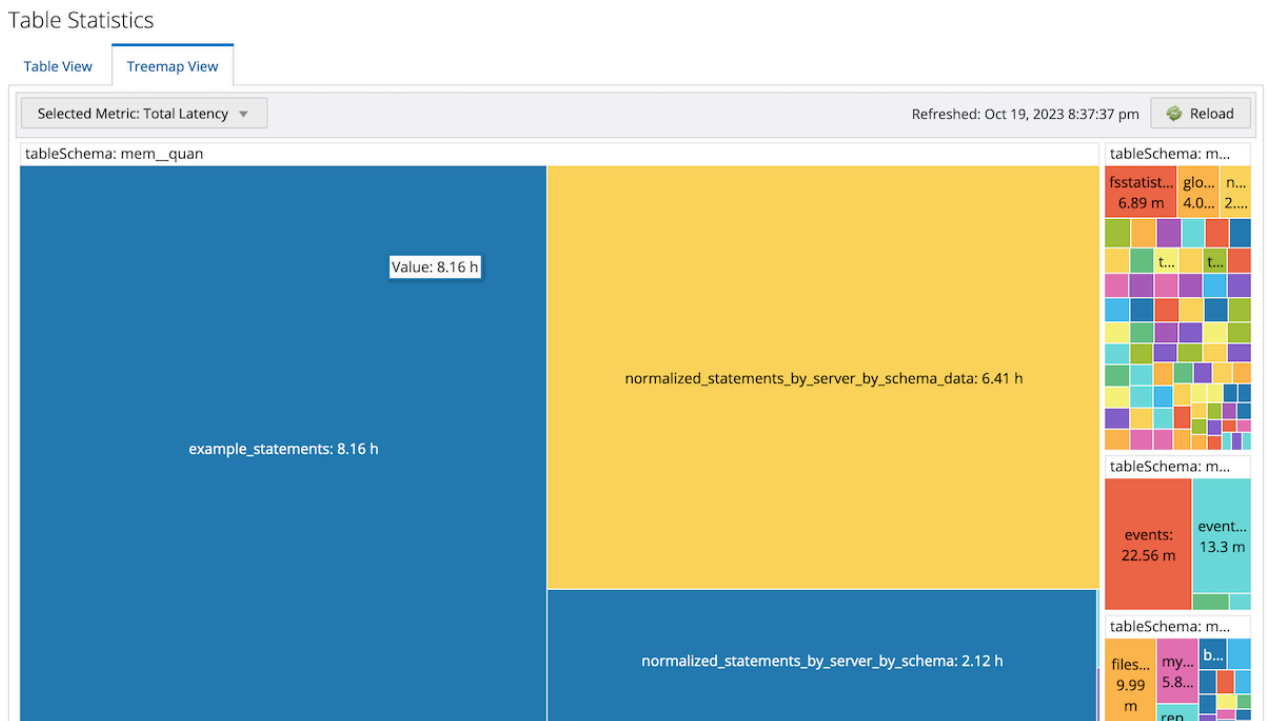
Table Schema	Table Name	Total Latency	Total Rows Fetched	Total Fetch Latency	Total Rows Inserted	Total Latency
mem_events	events	1.16 h	12744096918	56.72 m	62970	40
mem_events	event_chain_heads	21.92 m	165734718	11.46 m	6134	2.1
mem_events	events_subjects	2.09 m	27434352	1.96 m	105209	6.7
mem_events	subjects	1.35 m	18327320	1.35 m	38	27
mem_instruments	globaleventwaitsfilessummaryadvisor_fileioe...	25.48 m	1370405	31.98 s	1316796	6.2
mem_instruments	binlogcachedata	11.16 m	342456	9.13 s	329196	3.6
mem_instruments	fsstatistics	10.64 m	1732110	28.46 s	1645995	1.4
mem_instruments	networktrafficadvisor_networktraffic	10.3 m	1051659	1.99 m	669077	2.5
mem_instruments	diskiototaldata	9.93 m	243344	9.97 s	229199	2.6

Treemap View

The treemap displays the data in colored boxes. The larger the box, the larger the value.

Select the required metric from the drop-down menu.

Figure 30.8 Table Statistics Report - Treemap View



30.5 User Statistics

The **User Statistics** report is based on the user summary sys schema views. For more information, see [The user_summary and x\\$user_summary Views](#).



Note

This graph is displayed only if a MySQL instance is selected in the All Targets drop-down menu.

Table View

The table view lists the statistics in a standard table format. The data can be sorted by column.

Figure 30.9 User Statistics Report - Table View

User	Statements	Statement Latency	Statement Avg Latency	Table Scans	File IOs	File IO Latency
agent_general	554132	5.1 m	551.68 µs	32424	41961	1.22 s
agent_limited	27169	1.14 m	2.52 ms	20471	6560	10.89 ms
root	5472	12.26 s	2.24 ms	11	346	5.66 ms
background	0	0 ps	0 ps	0	7539	198.3 ms

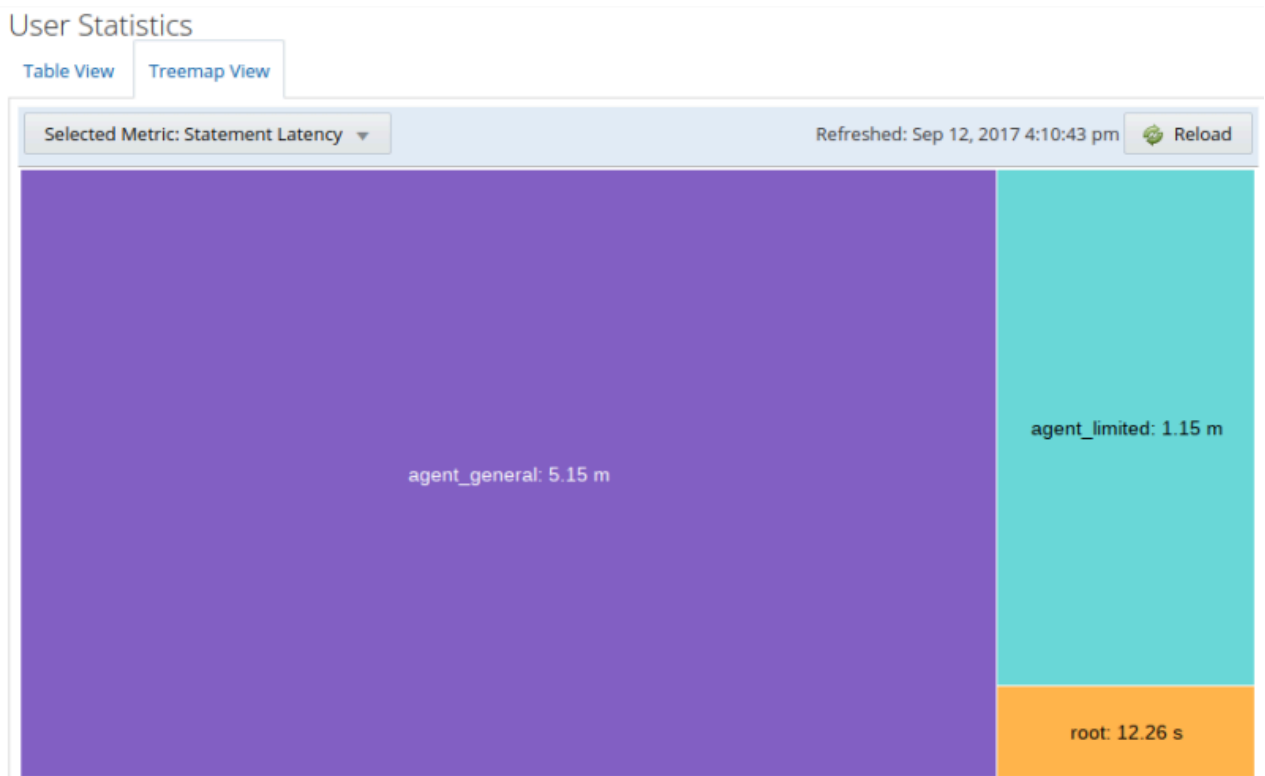
- Existing users who have never logged on to the instance are not listed in the results.
- The `mysql.sys` user is not listed in the results.
- The `background` user is the owner of the background threads running on the instance. Such as the storage engine threads, main server thread, and event scheduler.

Treemap View

The treemap displays the data in colored boxes. The larger the box, the larger the value.

Select the required metric from the drop-down menu.

Figure 30.10 User Statistics Report - Tree View



30.6 Memory Usage Report

The Memory Usage report summarizes memory usage, grouped by allocation type. This report is based on the `memory_global_by_current_bytes` sys view. For more information, see [The memory_global_by_current_bytes and x\\$memory_global_by_current_bytes Views](#).



Important

This instrumentation is disabled by default in MySQL 5.7, and enabled by default in MySQL 8.0. To enable in MySQL 5.7, add the following to your server configuration:

```
performance-schema-instrument='memory/%=ON'
```



Note

This report requires sys schema be installed on the selected MySQL instance. If sys is not installed, and the selected instance is compatible, you are prompted to install it.



Note

This graph is displayed only if a MySQL instance is selected in the **All Targets** drop-down menu.

Table View

The table view lists the memory usage in a standard table format. The data can be sorted by column.

Figure 30.11 Memory Usage Report - Table View

Memory Usage

Table View Treemap View

Type to filter Refreshed: Mar 29, 2018 9:25:38 am Reload

Show 10 entries Page 1 of 7 (1-10 of 66 items)

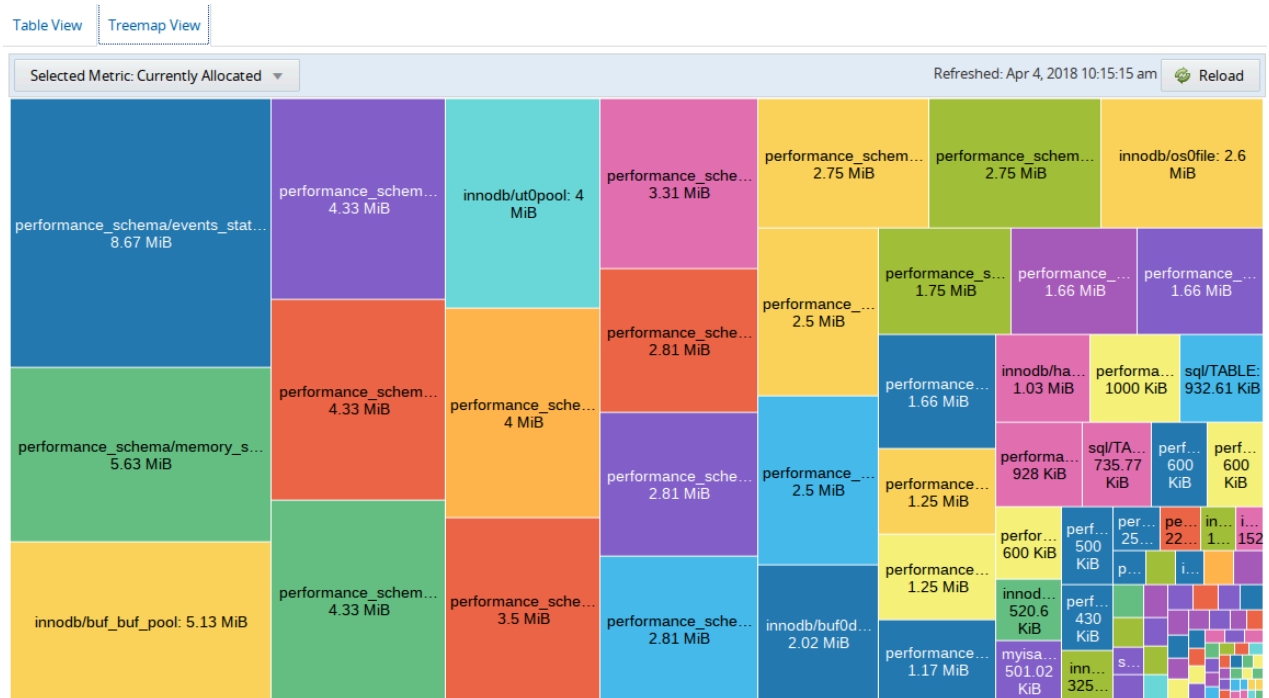
Memory Type	Current Allocations	Currently Allocated	Current Avg Allocation	Ma
performance_schema/file_instances	55	151.25 MiB	2.75 MiB	55
performance_schema/table_shares	28	112 MiB	4 MiB	28
performance_schema/rwlock_instances	111	13.88 MiB	128 KiB	111
performance_schema/events_statements_history_long	1	13.66 MiB	13.66 MiB	1
performance_schema/events_statements_summary_by_user_by_event_name	3	13 MiB	4.33 MiB	3
performance_schema/events_statements_summary_by_account_by_event_name	3	13 MiB	4.33 MiB	3
performance_schema/events_statements_summary_by_digest.tokens	1	9.77 MiB	9.77 MiB	1
performance_schema/events_statements_history_long.sqltext	1	9.77 MiB	9.77 MiB	1
performance_schema/events_statements_history_long.tokens	1	9.77 MiB	9.77 MiB	1
performance_schema/table_handles	1	9.06 MiB	9.06 MiB	1

Treemap View

The treemap displays the data in colored boxes. The larger the box, the larger the value.

Select the required metric from the drop-down menu.

Figure 30.12 Memory Usage Report - Tree View



30.7 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 [Buffer Pool](#)



Important

This report requires the `INFORMATION_SCHEMA.INNOODB_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 canceled and restarted. If you navigate away from the progress page, the report generation process is canceled.

5. The report is displayed.

30.8 NDB Cluster Memory Usage

The **NDB Cluster Memory Usage** report is based on the output from the `ndbinfo.memory_per_fragment` table. For more information, see [The ndbinfo memory_per_fragment Table](#).



Note

This graph is displayed only if an NDB Cluster is selected in the **Global Summaries** drop-down menu, and All Targets from the **All Targets** menu.

Table View

The table view lists the statistics in a standard table format. The data can be sorted by column.

Figure 30.13 NDB Cluster Memory Usage Report - Table View

NDB Cluster Memory

Table View Treemap View

Type to filter Refreshed: Sep 15, 2017 3:19:00 pm Reload

Show 10 entries Page 1 of 27 (1-10 of 264 items) K < 1 2 3 4 5 ... 27 > X

Schema Name	Table Name	Object Name	Type	Fragment Number	Node Id	Block Instance	Fixed Size Allocation	Fixed Size Eleme
world_ndb	city		User table	0	3	1	96 KIB	368 b
world_ndb	city		User table	3	6	1	96 KIB	7.64 KIB
world_ndb	city		User table	0	4	1	96 KIB	368 b
world_ndb	city		User table	1	5	1	96 KIB	3.95 KIB
world_ndb	city		User table	1	6	1	96 KIB	3.95 KIB
world_ndb	city		User table	2	3	1	96 KIB	3.23 KIB
world_ndb	city		User table	2	4	1	96 KIB	3.23 KIB
world_ndb	city		User table	3	5	1	96 KIB	7.64 KIB
ndbmemcache	containers		User table	2	4	1	32 KIB	31.8 KIB
ndbmemcache	cache_policies		User table	0	4	1	32 KIB	31.8 KIB

Treemap View

The treemap displays the data in colored boxes. The larger the box, the larger the value.

Select the required metric from the drop-down menu.

Figure 30.14 NDB Cluster Memory Usage Report - Tree View



Chapter 31 Customizing MySQL Enterprise Monitor

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

31.1 Creating Advisors and Rules

For common scenarios, reuse or edit the advisors and graphs provided by MySQL Enterprise Monitor. To create new advisors and graphs, open the **Advisors** page and select **Create Advisor** to create a custom advisor, or select **Import/Export** to create a graph.

31.1.1 Creating Advisors

Similar Advisors are grouped together in categories. To create a new Advisor, select **Create Advisor** on the **Advisors** page.



Note

Advisors have limited edit functionality, only threshold parameters and schedule type can be edited for an existing advisor.

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.

31.1.2 Overview of Graph Creation

Graphs are defined in XML files, then imported into MEM. The new custom graph is displayed in the **Graphing** category on the **Advisors** page.

For an example of how to create a graph, see [Section 31.1.9, “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.

31.1.3 Overview of Advisor Creation

To create a new Advisor, click the **Create Advisor** button on the Advisors page. To create an Advisor based on an existing one, select **Copy Advisor** menu item from the Advisor menu. You can edit any

Advisor element during the copying process, unlike editing an existing Advisor. To delete custom Advisors, select **Delete Advisor** from the Advisor menu.



Important

Unlike the expression-based advisors, it is not possible to copy one of the GUI-based advisors to use as the basis for a custom advisor.

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.

You can also edit an Advisor's expression. For more information on Advisor expressions, see [Chapter 19, Advisors](#). 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 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” advisor 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. MySQL Enterprise Monitor 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, System Variable, and Status 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.

31.1.4 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” advisor calculates the hit rate as follows:

```
100-((%Innodb_buffer_pool_reads% / %Innodb_buffer_pool_read_requests%)*100)
```

If the current value of `%Innodb_buffer_pool_reads%` is 4522 and the current value of `%Innodb_buffer_pool_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 advisor, 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 `Innodb_buffer_pool_reads` is `%Innodb_buffer_pool_reads%`.” to the [Advice](#) text box. When this is displayed on the screen, the value of `%Innodb_buffer_pool_reads%` is substituted into the text. If `%Innodb_buffer_pool_reads%` has a value of 4522, the message becomes “The current value of `Innodb_buffer_pool_reads` is 4522.”

31.1.5 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” advisor 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” advisor 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 advisor is not evaluated unless there is plenty of activity associated with the key cache.

31.1.6 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” advisor 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.

31.1.7 Wiki Format

When editing or defining an advisor, 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 31.1 Formatting with Wiki Markup

Markup	Description
<code>__bold__</code>	Boldface text
<code>~~italic~~</code>	Italicize text
<code>\n</code>	Create a line break
<code>\n\n</code>	Create a double line break

Markup	Description
<code>\\n</code>	Create a double line break
<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

For example, 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 \\n\\nSTART SLAVE; {'moreInfo:MySQL Manual: Replication
FAQ|http://dev.mysql.com/doc/refman/en/faqs-replication.html'}
```

is 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 \n\nSTART SLAVE;
<a href="http://dev.mysql.com/doc/refman/en/faqs-replication.html"
target="_blank" >MySQL Manual: Replication FAQ</a>
```

31.1.8 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 advisor 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”](#).

31.1.9 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 31.1.2, “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>Custom Graph 001</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>

  <displayprofile>
    <name>Disk Usage - All Filesystems</name>
    <type>Aggregate</type>
    <aggregate op="sum">
      <source>used</source>
      <source>total_size</source>
    </aggregate>
  </displayprofile>

  <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 notification is displayed.

This also creates a new Advisor with the same name as the new graph, 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 [Custom Graph 001](#) in the example above.

31.2 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.

31.2.1 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 31.2 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 31.2.2.1, "Precondition Queries" for more information.
<code>query</code>	The main query. For more information, see Section 31.2.2.2, "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 31.2.3, "Data Collection Attributes" for more information.

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

31.2.2 Queries

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

31.2.2.1 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.

31.2.2.2 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 31.2.3.2, “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.

31.2.2.3 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 31.1.7, “Wiki Format”](#) for more information on the supported wiki markup.

31.2.3 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 31.3 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. <ul style="list-style-type: none"> <code>true</code>: the attribute is a counter type. <code>false</code>: the attribute is not a counter type.
type	The attribute value type. Possible values are INTEGER, STRING or FLOAT.



Important

If an attribute type is incorrectly defined in the attribute definition, such as INTEGER instead of STRING, it is not possible to change the value in the `custom.xml` after the agent has started. This is because it is not possible for the agent to alter attribute types after they are defined. Attempting to change it in that manner results in an `InvalidValueForTypeException` error. To correct this, you must stop the agent, edit the type definition, rename the attribute, and restart the agent.

31.2.3.1 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>
```

31.2.3.2 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>
```

31.3 Event Notification Blackout Periods

During maintenance periods for database servers, you can suspend Event Handlers. 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 grayed 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
```

The URL is composed of the following components:

- Service Manager hostname: the URL of MySQL Enterprise Service Manager.
- `command`: the command to run. This can be one of the following:
 - `blackout`: applies or removes a `blackout_state` to the specified asset types.
 - `blackout_status`: requests the current blackout status of the specified asset.
- `blackout_state`: can be one of the following:
 - `true`: blackout the specified assets.
 - `false`: remove the blackout from the specified assets.

- Asset type: this can be one of the following:
 - `server_name`: the name and port number of the MySQL instance.
 - `server_uuid`: the UUID of the MySQL instance.
 - `group_name`: the name of the group. See [Chapter 18, Managing Groups of Instances](#) for more information.
 - `asset_type`: (only used for blackout of agents and `blackout_status` of agents) can be one of the following
 - `agent.Agent`: specifies the agent as the asset type. The command, or URL, must also contain the `agent_id` parameter.
 - `mysql.MysqlServer`: specifies the MySQL instance as the asset type. The command, or URL, must also contain the `id` parameter, with the value of the MySQL instance UUID.

Log on to the MySQL Enterprise Service Manager as a user with blackout privileges. For more information, see [Section 23.5, “MySQL Enterprise Monitor”](#). Use the username 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 grayed 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.

To blackout an agent, use the following URL:

```
https://localhost:18443/rest?command=blackout&agent_id=agentID&blackout_state=true
```

To blackout an agent and all the MySQL instances currently monitored by that agent, use the following URL:

```
https://localhost:18443/rest?command=blackout&agent_id=agentID&blackout_state=true&recursive=true
```

To check the blackout status of an agent, use the following URL:

```
https://localhost:18443/rest?command=blackout_status&id=agentID&asset_type=agent.Agent
```

To check the blackout status of a MySQL instance, use the following URL:

```
https://localhost:18443/rest?command=blackout_status&id=mysqlUUID&asset_type=mysql.MysqlServer
```

You can also check the blackout status of a MySQL instance using its server name. For example:

```
https://localhost:18443/rest?command=blackout_status&server_name=serverName
```

**Note**

It is not currently possible to check the blackout status of a group.

31.3.1 Scripting Blackouts

It is also possible to manage blackouts from the command line, or a script. The examples listed in this section all use `curl`. You can run the commands from the command line or add to a text file and save it as `*.sh` or `*.bat` depending on your platform.

**Note**

On Unix systems, use the `chmod +x blackout.sh` command to make the file executable.

The following blackout types are described:

- **Server Blackout:** describes how to apply or remove a blackout on a MySQL instance.
- **Group Blackout:** describes how to apply or remove a blackout on a MySQL Enterprise Monitor group.
- **Agent Blackout:** describes how to apply or remove a blackout on a MySQL Enterprise Monitor Agent and, recursively, on all MySQL instances monitored by the agent.
- **Blackout Status:** describes how to check the blackout status on the agent, group, or MySQL instance.

**Note**

Restarting MySQL Enterprise Service Manager or MySQL Enterprise Monitor Agent does not remove an applied blackout.

Server Blackout

The `blackout` command enables you to blackout individual MySQL instances.

To blackout an individual MySQL instance you must provide the following:

- `--user`: the username and password of the Service Manager user running the command. Ensure this user has the correct rights to perform blackouts.
- The URL of the Service Manager's rest API. For example: `"https://servicemanager:18443/rest"`
- `command=blackout`: the blackout command.
- One of the following:
 - `server_name=ServerName`: the name of the server to blackout.
 - `server_uuid=AlphanumericString`: the UUID of the MySQL instance to blackout.

To blackout a server using the `server_name`, use the following command:

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout" --data-urlencode "server_name=servername:3306"
--data-urlencode "blackout_state=true"
```

To blackout a server using the `server_uuid`, use the following command:

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout"
--data-urlencode "server_uuid=62e74f29-526a-11ba-77fb-0242xy112504"
--data-urlencode "blackout_state=true"
```

To remove the blackout, run the same command, with `"blackout_state=false"`. For example:

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout"
--data-urlencode "server_uuid=62e74f29-526a-11ba-77fb-0242xy112504"
--data-urlencode "blackout_state=false"
```

It is also possible to blackout all the MySQL instances monitored by a specific Agent. for more information, see [Agent Blackout](#).

Group Blackout

The `blackout` command enables you to blackout groups of MySQL instances. For more information on MySQL Enterprise Service Manager Groups, see [Chapter 18, Managing Groups of Instances](#).

To blackout a group of MySQL instances you must provide the following:

- `--user`: the username and password of the Service Manager user running the command. Ensure this user has the correct rights to perform blackouts.
- The URL of the Service Manager's rest API. For example: `"https://servicemanager:18443/rest"`
- `command=blackout`: the blackout command.
- `group_name=GroupName`: the name of the group to blackout.

The following command blackouts all MySQL instances in the group named `Group 1`.

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout" --data-urlencode "group_name=Group 1"
--data-urlencode "blackout_state=true"
```

To remove the blackout, run the same command, with `"blackout_state=false"`.

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout" --data-urlencode "group_name=Group 1"
--data-urlencode "blackout_state=false"
```

Agent Blackout

The command `blackout` enables you to blackout an Agent, or an Agent and all the MySQL instances it is currently monitoring.

To blackout an agent you must provide the following:

- `--user`: the username and password of the Service Manager user running the command. Ensure this user has the correct rights to perform blackouts.
- The URL of the Service Manager's rest API. For example: `"https://servicemanager:18443/rest"`
- `command=blackout`: the blackout command.
- `id=alphanumericString`: the id of the Agent to blackout.

To blackout an agent, run the following command:

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout"
--data-urlencode "agent_id=1e365f83-b703-4588-9472-877c90a2c49f"
--data-urlencode "blackout_state=true"
```

To blackout an Agent and all the MySQL instances it monitors, run the following command:

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout"
--data-urlencode "agent_id=1e365f83-b703-4588-9472-877c90a2c49f"
--data-urlencode "blackout_state=true" --data-urlencode "recursive=true"
```

Blackout Status

The command `blackout_status` enables you to check if an agent or MySQL instance is currently blacked out.

To check the blackout status you must provide the following:

- `--user`: the username and password of the Service Manager user running the command. Ensure this user has the correct rights to perform blackouts.
- The URL of the Service Manager's rest API. For example: `"https://servicemanager:18443/rest"`
- `command=blackout_status`: the blackout status command.
- `id=alphanumericString`: the id of the Agent or MySQL instance being checked.
- `asset_type`: one of the following:
 - `agent.Agent`: to check the blackout status of the Agent.
 - `mysql.MySqlServer`: to check the blackout status of the MySQL instance.

The following command checks the blackout status of an agent:

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout_status"
--data-urlencode "id=1e365f83-b703-4588-9472-877c90a2c49f"
--data-urlencode "asset_type=agent.Agent"
```

The following command checks the blackout status of a MySQL instance:

```
curl -G -k --user myadmin:mypassword "https://servicemanager:18443/rest"
--data-urlencode "command=blackout_status"
--data-urlencode "id=750cead2-441d-11ea-b3e7-dfdaa5ad712e"
--data-urlencode "asset_type=mysql.MySqlServer"
```

Part V Using the Query Analyzer

Table of Contents

32 Query Analyzer View	305
32.1 Providing Query Analyzer Data	305
32.1.1 Using the MySQL Performance Schema	305
32.2 Query Response Time index (QRTi)	307
32.3 Query Analyzer User Interface	308
32.4 Detailed Query Information	311
32.5 Query Analyzer Configuration View	314

Chapter 32 Query Analyzer View

The MySQL Query Analyzer enables you to monitor SQL statements executed on a MySQL server and displays the details of each query, number of executions and execution times. Similar queries with different literal values are combined for reporting purposes.

Query Analyzer collects information on the SQL statements MySQL client applications send to the MySQL server using the Performance Schema statement digests (MySQL Server 5.6.14 and above). Data can be gathered directly from MySQL Server without additional configuration, using a MySQL Enterprise Monitor Agent

For more information on the Query Analyzer user interface, see [Section 32.3, “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 32.3, “Query Analyzer User Interface”](#).

32.1 Providing Query Analyzer Data

The MySQL Query Analyzer retrieves data from the Performance Schema. The Performance Schema supplies the statistical information about the queries, execution times, result counts and other data to display and analyze on the Query Analyzer page.

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.

32.1.1 Using the MySQL Performance Schema

Query Analyzer data is automatically collected and displayed by monitoring MySQL Server 5.6.14 or greater, using the Performance Schema Statement Digests feature ([Performance Schema Statement Digests and Sampling](#)) added in MySQL 5.6.



Note

It is not possible to retrieve statement digest data from MySQL server versions prior to MySQL 5.6.14.

Collecting Query Analyzer data from Performance Schema provides data about how statements generate their result sets:

- 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

MySQL Enterprise Monitor Agent polls the [performance_schema.events_statements_summary_by_digest](#) table (every minute, by default) and continually computes 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

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.



Note

The maximum space available for digest computation is 1024 bytes by default; queries exceeding this length are truncated.

As of MySQL 5.7.8, and later, and 5.6.26, and later, this value can be changed at server startup by setting the `performance_schema_max_digest_length` system variable. In MySQL 5.6.24, 5.7.6, and 5.7.7, use `max_digest_length` instead. For MySQL 5.7 versions prior to 5.7.6, the value cannot be changed. Nor can it be changed for MySQL 5.6 versions prior to 5.6.24.

The `performance_schema.events_statements_summary_by_digest` table is a sized table in memory within the Performance Schema, and its size is auto-configured. To check the current size:

```
mysql> SHOW GLOBAL VARIABLES LIKE 'performance_schema_digests_size';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| performance_schema_digests_size | 5000  |
+-----+-----+
```

If your application executes more than this number of normalized statements, then it is possible that you may begin losing some statement instrumentation. You can monitor this situation with the `Performance_schema_digest_lost` 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 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.

If 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. Data is collected in this way from 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`.

32.2 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 32.1 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 percentage, yellow the acceptable percentage, and red the unacceptable percentage. 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.

32.3 Query Analyzer User Interface

To open the Query Analyzer, select Queries from the navigation menu.

Figure 32.1 Default Query Analyzer View

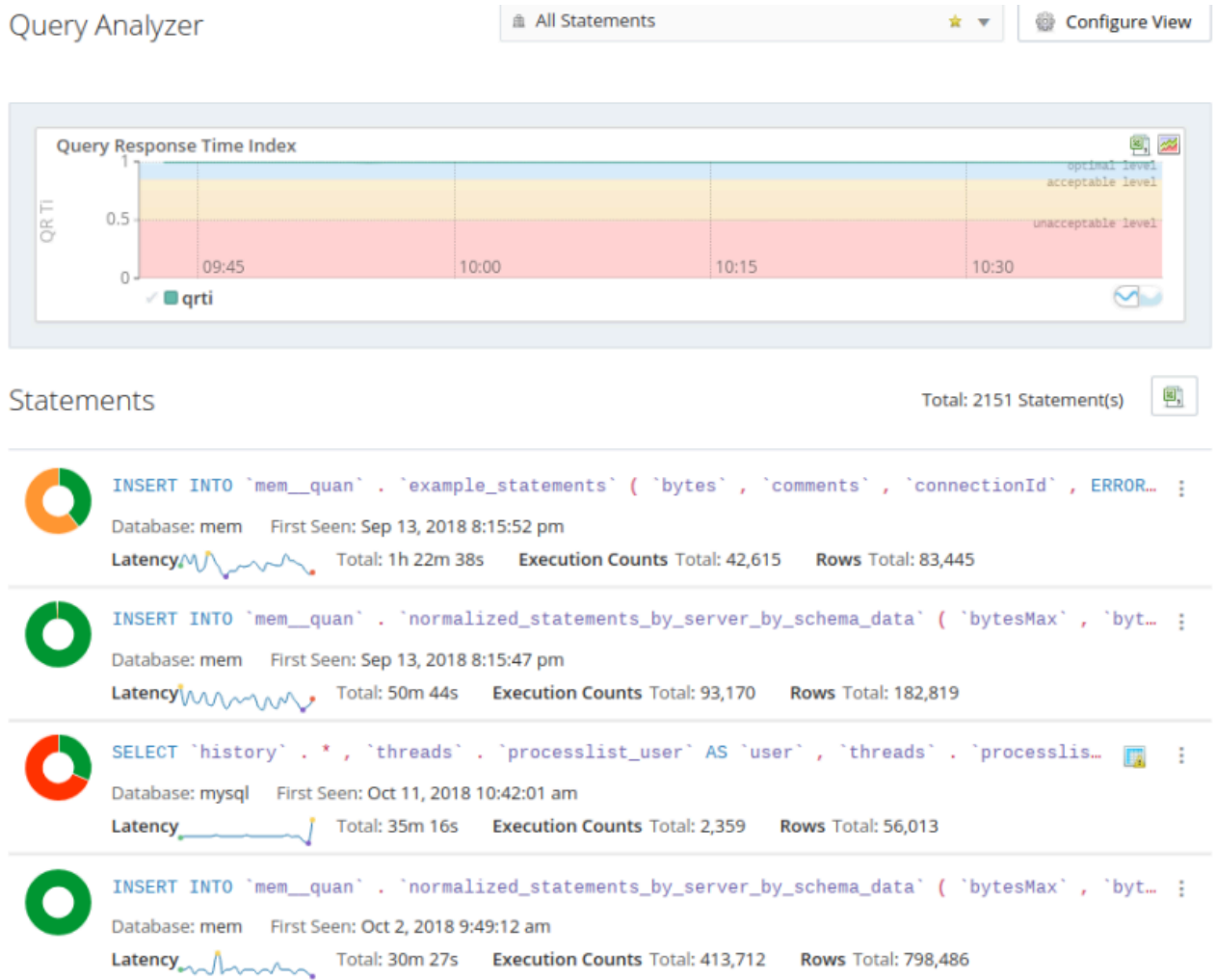


Table 32.2 Controls of the default Query Analyzer view

Control	Description
Filters	<p>Query Analyzer: contains the following default, system filters:</p> <ul style="list-style-type: none"> • Administration Statements: filters on statements of type GRANT, REVOKE, RESET, SET, SHOW, FLUSH, CACHE, KILL, and SHUTDOWN. • All Statements: (default) no filtering defined. All statements are displayed. • DDL Statements: filters on statements of type CREATE, DROP, ALTER, TRUNCATE, and RENAME. • DML Statements: filters on statements of type SELECT, INSERT, UPDATE, DELETE, REPLACE, CALL, LOAD, DO, and HANDLER. • Prepared Statements: filters on statements of type PREPARE, EXECUTE, and DEALLOCATE. • Replication Statements: filters on statements of type START, STOP, RESET, and CHANGE

Control	Description
	<ul style="list-style-type: none"> • Statements with Errors: filters on the advanced filter options of Total Errors > 0. • Statements with Full Table Scans: filters on the advanced filter options of Table Scan notices and Total Table Scans > 0. • Statements with Max Exec Time Over 1 Second: filters on the advanced filter options of Max Exec Time > 1. • Statements with Temporary Tables: filters on the advanced filter options of Total Temporary Tables > 0. • Statements with Temporary Tables on Disk: filters on the advanced filter options of Total Temporary Disk Tables > 0 • Statements with Warnings: filters on the advanced filter options of Total Warnings > 0 • Table Maintenance Statements: filters on statements of type OPTIMIZE, ANALYZE, CHECK, REPAIR, and CHECKSUM. • Transactional and Locking Statements: filters on statements of type BEGIN, COMMIT, ROLLBACK, SAVEPOINT, RELEASE, LOCK, and UNLOCK. • Customized versions of the system filters are created using the name of the filter with a -clone suffix.
Configure View	<p>Opens the configuration view.</p> <p>The Configure View enables you to customize the data displayed on the Query Analyzer view. For more information, see Section 32.5, “Query Analyzer Configuration View”.</p>
Statements	<p>The statements panel displays the statement data. The data displayed is configured in the Data View section of the Configuration View.</p> <p>For more information, see Data View.</p> <p>By default, the Statements view displays the following:</p> <ul style="list-style-type: none"> • QRTi pie chart: visual representation of the Query Response Time index. Hover the cursor over the pie chart to see a summary of the Optimal, Acceptable, and Unacceptable QRTi percentages. • Query: displays the normalized query. Hover the cursor over the statement to see the full version. Or click the query to open the Details page. • Database: the name of the database on which the query was executed. • First Seen: the time and data at which this query was first seen on the database. • Latency: sparkline graph displaying the latency of the query. Hover the cursor over the graph to display a tooltip containing the summary of the latency. • Total: total amount of time spent on this query. • Execution Counts: number of times this query was executed.

Control	Description
	<ul style="list-style-type: none"> • Rows: number of rows affected by this query. <p>All elements of the statement view are configurable, except for the statement itself, which is always displayed. To display the full statement, hover the cursor over the statement line. The full statement is displayed in a pop-up. To view the details of the statement, click the statement, or the more options button on the right of the statement's panel.</p> <p>For more information on the Details page, see Section 32.4, “Detailed Query Information”.</p>

32.4 Detailed Query Information

Click on an individual query to see more detailed information about the individual query in the **Details** view.

For more information on Normalization and Statement Digests, see [Performance Schema Statement Digests and Sampling](#).



Note

If you change the context, by selecting another MySQL instance from the **All Targets** menu, and the selected instance does not contain the statement digest of the originally selected query, an error is displayed.

The details page contains the following elements:

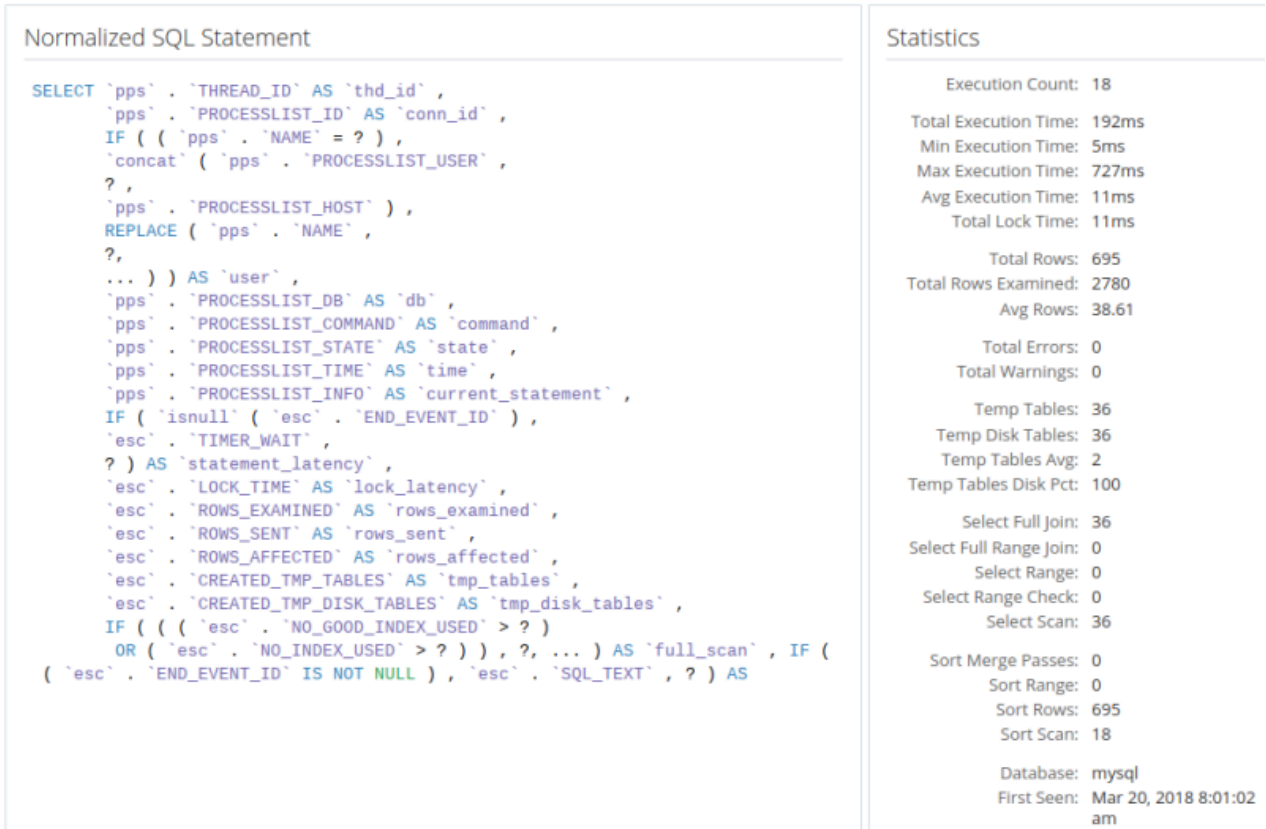
- **Statement Digest** field: lists the truncated identifier of the statement. To see the full version, click **Show**. To copy the statement digest to the clipboard, click **Copy**.
- Date and time range: displays the statement's time range, as selected on the **Query Analyzer** page.
- **Normalized SQL Statement:** displays the selected, normalized SQL statement.
- **Statistics:** displays the relevant statistics for the SQL statement. Number of executions, Total Execution time, and so on.
- **Graphs:** displays the average execution time, number of executions, and number of rows affected by the statement within the selected time range.
- **Example Statements:** (Displayed only if Example Query is enabled) distribution chart of the statement by **Execution Time**, in milliseconds, and time and date, within the selected time range.
- **Example SQL Statements:** (Displayed only if Example Query is enabled)

Normalized SQL Statement and Statistics

The **Normalized SQL Statement** displays the normalized version of the SQL statement, while the **Statistics** displays useful information on the execution of that statement.

Figure 32.2 Normalized Queries and Statistics

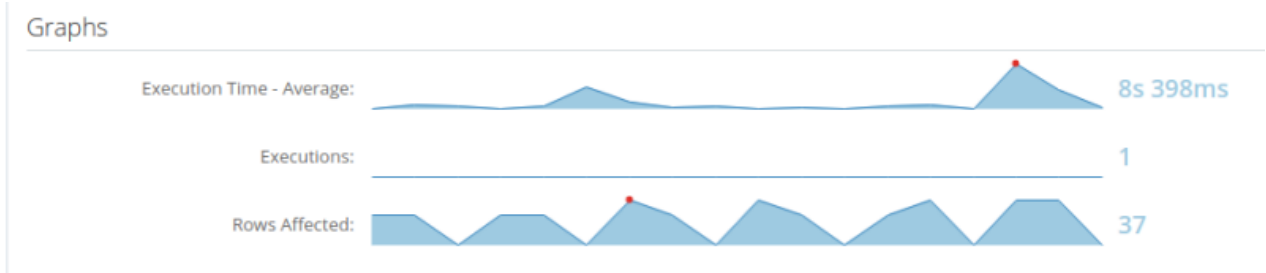
From Apr 10, 2018 2:53:27 pm to Apr 10, 2018 3:23:27 pm.



Graphs

The statement graph displays the average execution time, number of executions, and the number of rows affected.

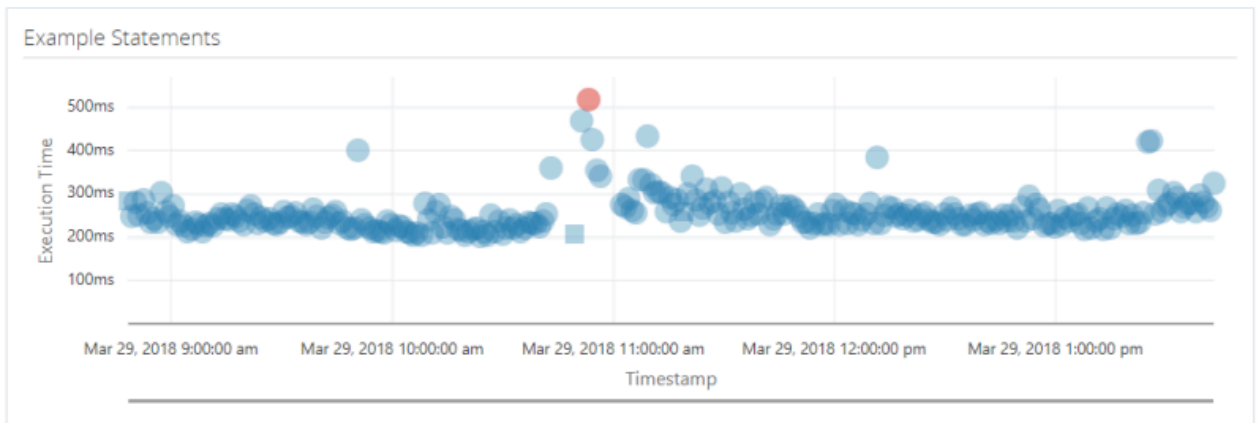
Figure 32.3 Statement Detail Graphs



Example Statements, Details, and EXPLAIN

The Example Statements distribution chart graphically represents the statement by execution time and time and date. Each point on the chart represents a specific example statement execution time.

Figure 32.4 Example Statements Distribution Chart



Statements with an associated EXPLAIN plan are represented by a circle in the Distribution Chart. Squares have no EXPLAIN plan. The red circle or square is the SQL statement with the longest execution time for the specified period. Clicking on a circle or square displays its query text in the original form, execution details, and its EXPLAIN plan, if EXPLAIN is enabled. Selecting one of the points loads the data for that point in the Example details frames:

Figure 32.5 Example Statement and Details

Example SQL Statement	Example Details
<pre> SELECT history.*, threads.processlist_user AS `user`, threads.processlist_host AS `hostFrom` FROM (SELECT statement_history.digest AS `digest`, sql_text AS `text`, current_schema AS `schema`, thread_id AS `connectionId`, IFNULL(statement_history.timer_wait * 0.000001, 0) AS `execTime`, errors AS `errors`, warnings AS `warnings`, rows_sent AS `rows`, no_index_used AS `noIndexUsed`, no_good_index_used AS `noGoodIndexUsed`, 1520236477000 + (timer_end DIV 1000000000) AS exampleTimestamp FROM performance_schema.events_statements_history_long AS statement_history JOIN (SELECT IFNULL(MAX(timer_wait), 0) AS timer_wait, digest, current_schema AS `schema` FROM performance_schema.events_statements_history_long -- filter by the exampleTimestamp per row WHEN finding the maxes, -- AS we only want rows IN our range considere d WHERE 1520236477000 + (timer_end DIV 1000000000) > 1522317078009 GROUP BY digest, current_schema ORDER BY NULL) AS max_times ON (max_times.timer_wait = statement_history.timer_wait AND max_times.digest = statement_history.digest AND max_times.schema <=> statement_history.current_schema)) AS history LEFT OUTER JOIN performance_schema.threads AS threads ON history.connectionId = threads.thread_id </pre>	<pre> Date: Mar 29, 2018 10:53:19 am Execution Time: 518ms Thread ID: 126 From Host: localhost User: mem MySQL Instance: </pre>

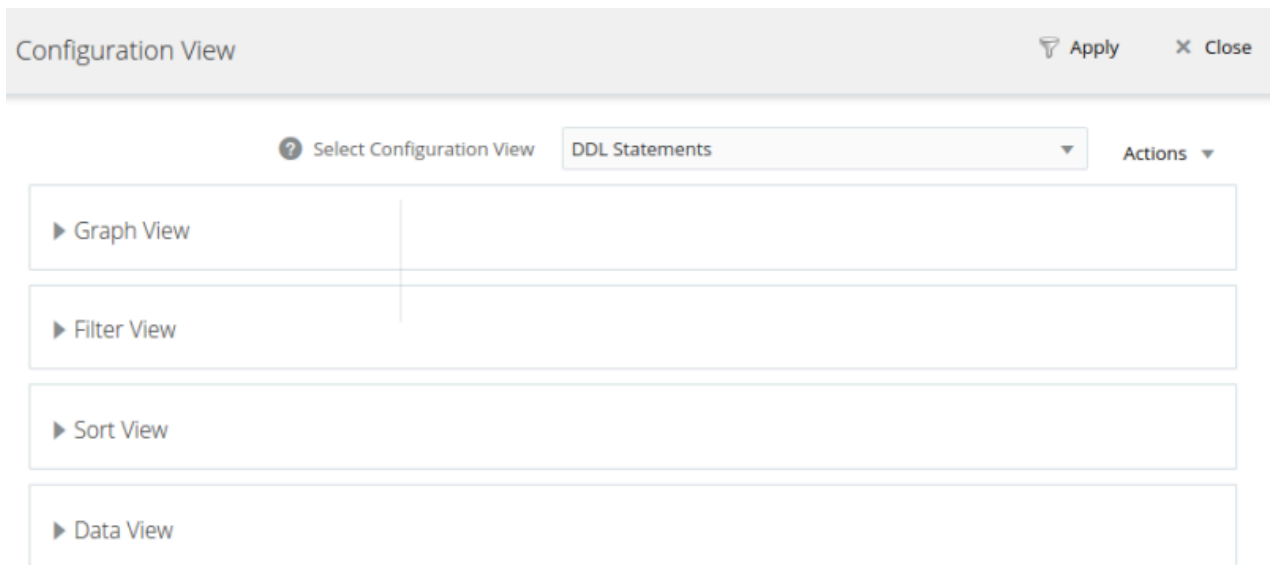
Figure 32.6 Example EXPLAIN

Example EXPLAIN

Id	select_type	table	type	possible_keys	key	key_len	ref
1	PRIMARY	statement_history	ALL			0	
1	PRIMARY	threads	ALL			0	
1	PRIMARY	<derived3>	ref	<auto_key0>	<auto_key0>	303	performance_schema.statement_history.TIME
3	DERIVED	events_statements_history_long	ALL			0	

32.5 Query Analyzer Configuration View

The **Configuration View** enables you to customize the data displayed on the **Query Analyzer** view.

Figure 32.7 Configuration View

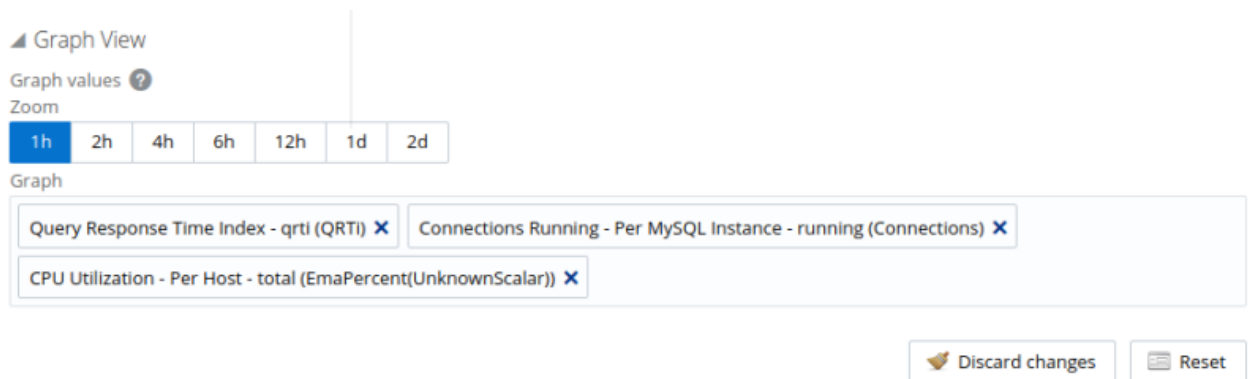
Graph View Selector

Enables you to select the graphs displayed on the Query Analyzer view and the time range of the selected graphs.

To select a time range for the graphs, select a value from the **Zoom** section. Possible values range from one hour to two days.

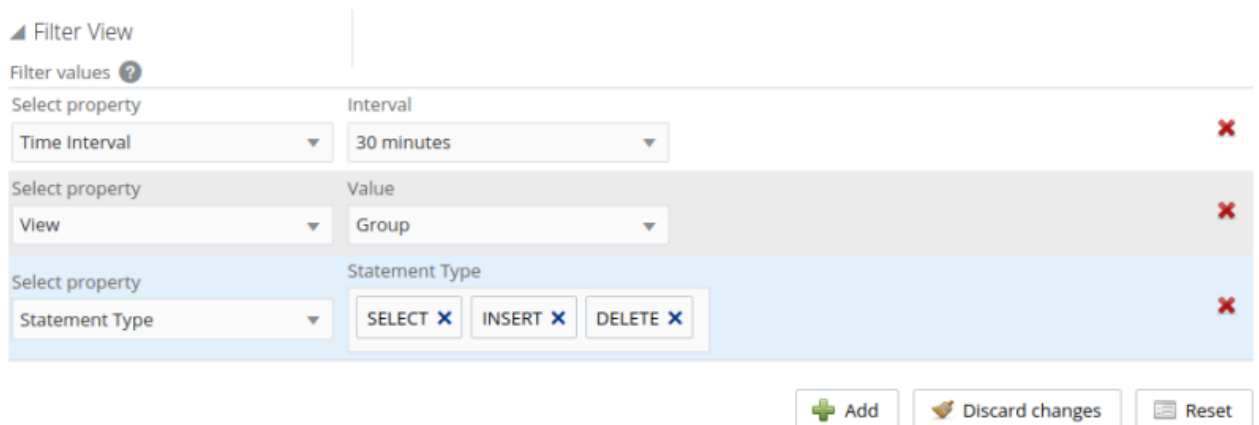
To select one or more graphs to display on the Query Analyzer view, click in the Graph selection box and select the required graphs from the available options.

Figure 32.8 Graph View



Filter View

Figure 32.9 Filter View



The following are the possible filter options:

- Column:** enables 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 on all queries that return more than 100 rows on average, set the **Column** to [Average Rows](#), the **Operator** to `>=`, and the **Value** to 100.
- Database:** 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](#).
- Notices:** Filters on the notices, enabling 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.
- 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.).
- The **Time From/To:** enables you to select a time range for filtering. Only queries executed within the displayed time period are displayed, using the **Hours** and **Minutes** pop-up), or whether the selection should be based on a time period (**From/To**).

Using this property you can display only the queries executed 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.



Note

It is not possible to use both **Time Interval** and **Time From\To** together. You must use one or the other.

- **Time Interval:** filters queries within the given time period from the point the graph was updated. For example, if you select 30 minutes, the queries displayed are those captured within the last 30 minutes. If you updated the display at 14:00, the queries displayed are those captured between 13:30 and 14:00. Possible values are between 15 minutes and 2 years.



Note

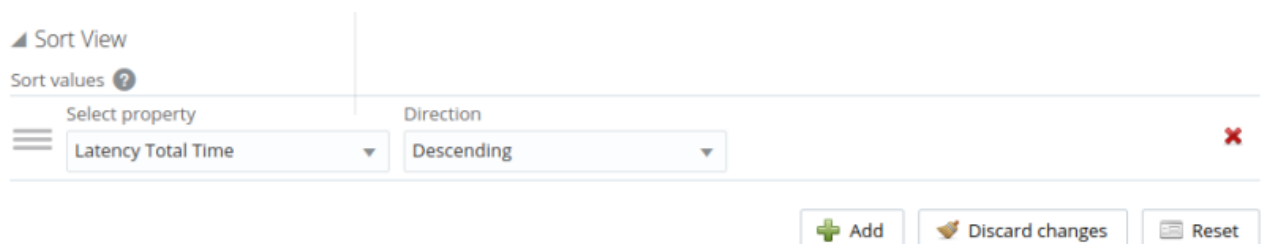
It is not possible to use both **Time Interval** and **Time From\To** together. You must use one or the other.

- **View:** 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 is selected, the statements can be expanded to list the individual servers on which they were executed.

Sort View

The Sort view enables you to specify how the data is ordered in the Statements view.

Figure 32.10 Sort View



- **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.

- **Errors:** 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.
- **Execution Counts:** The number of times that the query has been executed.
- **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.
- **Instance:** the name of the MySQL instance.
- **Latency:** 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).

Latency is subdivided into the following groupings:

- **Latency Total:** the cumulative execution time for all the executions of this query.
- **Latency Maximum:** the maximum execution time for an execution of this query.
- **Latency Average:** 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.
- **No Good Index Used:**
- **No Index Used:**
- **Rows:** The rows returned by the query. This is sub-divided into the following groupings:
 - **Rows Total:** The sum total number of rows returned by all executions of the query.
 - **Rows Average:** The average number of rows returned by all executions of the query.
 - **Rows Examined:** The average number of rows returned by all executions of the query.
- **Select Type**
 - **Select Type Full Join:** number of joins performing table scans because they do not use indexes.
 - **Select Type Full Range Join:** the number of joins using a range search on a reference table.
 - **Select Type Range:** the number of joins using ranges on the first table.
 - **Select Type Range Check:** the number of joins without keys that check for key usage after each row.
 - **Select Type Scan:** the number of joins performing a full scan of the first table.
- **Sorting**
 - **Sorting Merge Passes:** the number of merge passes the sort algorithm has performed.
 - **Sorting Range:** the number of sorts performed using ranges.

- **Sorting Rows:** the number of sorted rows.
- **Sorting Scan:** the number of sorts performed by scanning the table.
- **Statement:** 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.



Note

If the selected context is a cluster, the queries for the cluster's SQL nodes are displayed either for the specific node or for the cluster as an aggregate. If other node types are selected, the aggregate is displayed.

- **Temporary Tables**
 - **Temporary Tables Average:** the average number of internal temporary tables created per occurrence of the statement.
 - **Temporary Tables Disk:** the total number of internal, on-disk temporary tables created by occurrences of the statement.
 - **Temporary Tables Disk % :** the percentage of internal in-memory temporary tables that were converted to on-disk tables.
 - **Temporary Tables Total:** the total number of internal in-memory temporary tables created by occurrences of the statement
- **Time:** the time at which the statements were executed.
- **Warnings:** the number of warnings generated by a statement.

Data View

The **Data View** configures the elements displayed in the entries of the **Statement** view.

Figure 32.11 Data View

▲ Data View

Display values ?

Select property			
QRTI graph			✘
Select property			
Instance Name			✘
Select property	Items		
Latency	History Graph ✘ Total Time ✘		✘
Select property	Items		
Execution Counts	Total ✘		✘
Select property	Items		
Information Icons	Notices ✘ Error ✘ Warning ✘		✘
Select property	Items		
Rows	Total ✘		✘
Select property			
Database			✘
Select property			
First seen			✘

The possible properties are as follows:

- **Database:** the name of the database on which the statement was executed.
- **Execution Counts:** the number of times the statement was executed.
- **First Seen:** the time and date the statement was first seen.
- **Information Icons:** select which information icons you want displayed on the statement view. These icons are displayed on the right-hand side of the statement. Possible choices are: **Notices**, **Errors**, or **Warnings**.
- **Instance Name:** the name of the MySQL instance.
- **Latency:** the execution times for the statements. The following latency settings are possible:
 - **History Graph:** adds a latency graph to the statement data. This graphs the latency high, low, and average history.
 - **Total Time:** the total time taken by all executions of this statement.
 - **Maximum Time:** the maximum time taken for an execution of this statement.
 - **Average Time:** the average execution time for this statement.
 - **Lock Time:** the time spent waiting for table locks caused by this statement.

- **QRTi Graph:** adds a Query Response Time index pie-chart to the left side of the statement. The pie-chart graphs the Optimal, Acceptable, and Unacceptable percentages for the statement.
- **Rows:** the number of rows returned. The following are the possible values:
 - **Total:** the sum total of rows returned by executions of this statement.
 - **Examined:** the total number of rows read by this statement.
 - **Average:** the average number of rows returned by this statement.
- **Select Type:** the following are the possible values:
 - **Full Join:** the number of joins performing table scans because they do not use indexes.
 - **Full Range:** the number of joins using a range search on a reference table.
 - **Range:** the number of joins using ranges on the first table.
 - **Range Check:** the number of joins without keys that check for key usage after each row.
 - **Scan:** the number of joins performing a full scan of the first table.
- **Sorting:** the following are the possible values:
 - **Merge Passes:** the number of merge passes the sort algorithm has performed.
 - **Range:** the number of sorts performed using ranges.
 - **Rows:** the number of sorted rows.
 - **Scan:** the number of sorts performed by scanning the table.
- **Temporary Tables:** the following values are possible:
 - **Average:** the average number of internal temporary tables created per occurrence of the statement.
 - **Disk:** the total number of internal, on-disk temporary tables created by occurrences of the statement.
 - **Disk % :** the percentage of internal in-memory temporary tables that were converted to on-disk tables.
 - **Total:** the total number of internal in-memory temporary tables created by occurrences of the statement

Customizing Filters

Filters can be created, or existing filters customized, in the **Query Analyzer Configuration View**.

To create a user filter, you can either create a filter and save it by clicking **Save as...** menu item, or create a new filter by clicking **New**, defining your filter criteria, and clicking **Save as....**

You can also create a filter by using an existing filter as a template. Select the filter and make your changes. If you create a named filter based on an existing filter, the **-clone** is appended to the name when you edit the new filter. The name can be edited as required.

System filters are listed with a padlock on the left side of their name. These cannot be edited, but can be used as templates for new filters.

To set a filter as the default, select it in the drop-down list, and select **Set as Default** from the adjacent drop-down menu. A star is displayed next to the default filter.



Note

If you upgraded from a previous version, and used default filters in that version, your filters are migrated and renamed [User Default](#).

Part VI Appendices

Table of Contents

A MySQL Enterprise Monitor Component Reference	327
A.1 MySQL Enterprise Service Manager Reference	327
A.1.1 MySQL Enterprise Service Manager Log Files	327
A.1.2 The Management Information Base (MIB) File	327
A.1.3 The <code>config.properties</code> file	328
A.2 MySQL Enterprise Monitor Agent Reference	331
A.2.1 Agent Log Files	331
B Managing the Inventory	333
B.1 The Inventory Page	333
B.2 Using the Inventory Page	334
C MySQL Enterprise Monitor Frequently Asked Questions	335
D MySQL Enterprise Monitor Support	341
D.1 Diagnostics Report	341

Appendix A MySQL Enterprise Monitor Component Reference

Table of Contents

A.1 MySQL Enterprise Service Manager Reference	327
A.1.1 MySQL Enterprise Service Manager Log Files	327
A.1.2 The Management Information Base (MIB) File	327
A.1.3 The <code>config.properties</code> file	328
A.2 MySQL Enterprise Monitor Agent Reference	331
A.2.1 Agent Log Files	331

A.1 MySQL Enterprise Service Manager Reference

A.1.1 MySQL Enterprise Service Manager Log Files

This section shows the location of the log files associated with the various components that make up MySQL Enterprise Service Manager. These files can prove useful for debugging purposes.

All log files except `catalina-daemon.out` are rotated to ensure they do not grow beyond 10MB in size. The `catalina-daemon.out` Apache Tomcat log file is rotated (as `catalina.yyyy-mm-dd.log`) when Apache Tomcat is restarted, which happens when MySQL Enterprise Service Manager is restarted.

Table A.1 MySQL Enterprise Service Manager Log File Locations

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

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. The following table lists the location of this file per operating system:

Table A.2 Default Location of MIB Files

Operating System	File Location
Microsoft Windows	<code>C:\Program Files\MySQL\Enterprise\Monitor\support-files\MONITOR.MIB</code>
UNIX/Linux	<code>/opt/mysql/enterprise/monitor/support-files/MONITOR.MIB</code>
macOS	<code>/Applications/mysql/enterprise/monitor/support-files/MONITOR.MIB</code>

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 Default Location of `config.properties` File

Operating System	File Location
Microsoft Windows	<code>..\Monitor\apache-tomcat\webapps\ROOT\WEB-INF</code>
UNIX/Linux	<code>../monitor/apache-tomcat/webapps/ROOT/WEB-INF</code>
macOS	<code>../monitor/apache-tomcat/webapps/ROOT/WEB-INF</code>

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

The following examples shows a generated `config.properties` file:

```
#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-logging/configuration.html>.

Table A.4 Optional `config.properties` Values

Property Name	Property Type	Default
<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.minEvictable</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	30000 seconds
<code>dbpool.ui.minEvictableIdleTimeMillis</code>	String	15 seconds
<code>dbpool.ui.minIdle</code>	Integer	0
<code>dbpool.ui.timeBetweenEvictionRunsMillis</code>	String	5000 milliseconds

- `dbpool.ui.initialSize(5)`

Type	Integer
Default Value	5

The initial number of connections that are created when the pool is started.

- `dbpool.ui.maxActive(15)`

Type	Integer
Default Value	15

The maximum number of active connections that can be allocated from this pool at the same time.

- `dbpool.ui.minIdle(0)`

Type	Integer
Default Value	0

The minimum number of established connections that should be kept in the pool at all times. Default value is derived from `???`.

- `dbpool.ui.maxIdle(5)`

Type	Integer
Default Value	5

The maximum number of connections kept in the pool at all times. Idle connections are checked periodically, if enabled, and connections idle for longer than `minEvictableIdleTimeMillis` are released.

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

Type	String
Default Value	30000 seconds

The maximum number of milliseconds the pool waits for a connection to be returned before throwing an exception. Set to `-1` to wait indefinitely.

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

Type	String
Default Value	5000 milliseconds

The number of milliseconds to sleep between runs of the idle connection validation or cleaner thread. Do not set this value to less than 1 second. It specifies how often checks are performed for idle connections, and how often idle connections are validated.

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

Type	String
Default Value	15 seconds

The minimum amount of time an object may sit idle in the pool before it is eligible for eviction.

- `dbpool.default.initialSize(20)`

Type	Integer
Default Value	20

The initial number of connections created when the pool is started.

- `dbpool.default.maxActive(70)`

Type	Integer
Default Value	70

The maximum number of active connections allocated from this pool at the same time.

- `dbpool.default.minIdle(0)`

Type	Integer
Default Value	0

The minimum number of established connections kept in the pool at all times.

- `dbpool.default.maxIdle(5)`

Type	Integer
Default Value	20

The maximum number of connections kept in the pool at all times.

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

Type	String
Default Value	30 seconds

The maximum number of milliseconds the pool waits for a connection to be returned before throwing an exception.

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

Type	String
Default Value	5 seconds

The number of milliseconds to sleep between runs of the idle connection validation or cleaner thread. Do not set this value to less than 1 second. It specifies how often checks are performed for idle connections, and how often idle connections are validated.

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

Type	String
Default Value	15 seconds

The minimum amount of time an object may sit idle in the pool before it is eligible for eviction.

A.2 MySQL Enterprise Monitor Agent Reference

MySQL Enterprise Monitor Agent provides all of the information from a monitored MySQL server to MySQL Enterprise Service Manager.

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/`
- macOS Path: `/Applications/mysql/enterprise/agent/logs/`

The log files are managed with log4j, which is configured using `log4j2.xml`. The Agent watches for changes every 30 seconds, and updates MySQL Enterprise Monitor accordingly. The default file location:

- Windows Path: `C:\Program Files\MySQL\Enterprise\Agent\etc\log4j2.xml`
- Linux Path: `/opt/mysql/enterprise/agent/etc/log4j2.xml`
- macOS Path: `/Applications/mysql/enterprise/agent/etc/log4j2.xml`

The maximum size of a log file may be limited to 2GB. If MySQL Enterprise Monitor Agent cannot add information to the configured log file, 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:

```
<Policies>  
  <SizeBasedTriggeringPolicy size="10 MB">  
</Policies>  
<DefaultRolloverStrategy max="10" />
```

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	333
B.2 Using the Inventory Page	334

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 file systems 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 <https://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](#): Using MySQL Enterprise Service Manager 3.2 with older agents, why am I seeing warning messages like the following in the agent logs?

```
WARN [HttpPollingJobFetcher-thread-0:com.mysql.ertools.agent.ManagedOrder]
bean order property CoalescedCollectionRunner.collections is a collection
and contains a class not available on the agent. This element of the collection
will be ignored. Upgrading the agent to a version at least the same as the service
manager may solve this problem.
java.lang.RuntimeException: java.lang.ClassNotFoundException:
com.mysql.ertools.inventory.model.mysql.XXXXXXX
```

- [C.3](#): In 2.3, the `agent-mgmt-hostname` contained the string "heartbeat" as the URLs path. Did this change?

-
- **C.4:** How do I change the name of a server?
 - **C.5:** Does Query Analyzer work with all versions of MySQL and the MySQL Client Libraries?
 - **C.6:** 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.7:** 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: Using MySQL Enterprise Service Manager 3.2 with older agents, why am I seeing warning messages like the following in the agent logs?

```
WARN [HttpPollingJobFetcher-thread-0:com.mysql.ertools.agent.ManagedOrder]
bean order property CoalescedCollectionRunner.collections is a collection
and contains a class not available on the agent. This element of the collection
will be ignored. Upgrading the agent to a version at least the same as the service
manager may solve this problem.
java.lang.RuntimeException: java.lang.ClassNotFoundException:
com.mysql.ertools.inventory.model.mysql.XXXXXXX
```

MySQL Enterprise Monitor 3.2 introduced support for multi-source replication. Some of the functionality introduced in the 3.2 agents is not available in the earlier agents. `XXXXXX` represents one of the classes unavailable in the earlier agent.

These warnings do not imply anything is wrong with your installation or any data is missing from the collections. However, it is strongly recommended that you upgrade your agents to the same version as your MySQL Enterprise Service Manager.

C.3: 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.4: 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.5: 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.6: 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.7: 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. 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.

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 if the MySQL Query Analyzer is 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 handler prepared statements?
 - [C.11](#): 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 32.1, “Providing Query Analyzer Data”](#).

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

Using MySQL 5.6, or higher, with Performance Schema enabled, there is no appreciable overhead.

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

For the complete features and benefits, see [MySQL Enterprise Monitor Features and Benefits](#).

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 handler prepared statements?

At this time, the query analyzer does not track server-side prepared statements. However the default configurations for most client-side libraries for MySQL don't use them, they emulate them client-side, and those will be tracked by the query analyzer.

C.11: Will the Query Analyzer work without any special setup?

With MySQL Server 5.6.14 and greater, Query Analyzer data is automatically (by default) collected and displayed using the Performance Schema Statement Digests MySQL Server feature. If you are monitoring an earlier MySQL Server version, then you can continue to use alternative methods of providing query data to the Query Analyzer.

For information about the different methods of retrieving query data, see [Section 32.1, "Providing Query Analyzer Data"](#).

Appendix D MySQL Enterprise Monitor Support

Table of Contents

D.1 Diagnostics Report	341
------------------------------	-----

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 **Help** menu. The information is provided as a time stamped Zip file (such as `support-20160115T2238.zip`). 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`.

The Diagnostic Report is useful for debugging MySQL Enterprise Service Manager and 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-daemon.out`: A Tomcat log file.
- `com.mysql.etoools.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`: 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.

- `Replication n.json`: The MySQL server replication structure, where n is the number of the replication topology.
- `root.csv`: A copy of your main MySQL Enterprise Monitor log file.
- `server.props`: A copy of your server properties.

MySQL Enterprise Monitor Glossary

These terms are commonly used in information about MySQL Enterprise Monitor.

A

access control

MySQL Enterprise Monitor Access Control enables you to manage asset visibility, application administration, data access, roles, and users.

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

See Also [role](#).

advisor

Advisors filter and evaluate the information collected by the Monitoring Agents and present it to the Events page when defined thresholds are breached. Advisors can be in one or more Categories.

The MySQL Enterprise Monitor product comes with a number of default advisors. Based on your organization's particular needs, you may create or adapt custom advisors.

See Also [custom](#), [notification](#), [schedule](#), [standard](#).

advisor category

A set of Advisors that are designed to enforce MySQL best practices for that specific category. For example, there are predefined advisor categories such as [Administration](#), [Availability](#), [Cluster](#), [Replication](#) and [Security](#).

See Also [advisor](#).

Agent

The MySQL Enterprise Monitor component that actively collects data from a host and its MySQL server **instances**. The data is transmitted to the **Service Manager**.

See Also [instance](#), [Service Manager](#).

Apache

The web server that runs the **Tomcat** servlet container that powers the **UI**. The MySQL Enterprise Monitor **Service Manager** installation includes the Apache web server.

See Also [Service Manager](#), [Tomcat](#).

asset

A component that the MySQL Enterprise Monitor product monitors, such as a MySQL server **instance**, or a CPU or file system within a server machine. Contrast with **metric**, which is a property within the component that is measured.

See Also [instance](#), [metric](#).

Asset Selector

The Groups shown in the **Asset Selector** are the ones you create and manage (on the [MySQL Instances Dashboard](#), or during Agent installation) and those that are discovered automatically from monitoring replication topologies. It is on the left side of many UI pages that controls content in the main-body of the page.

See Also [asset](#), [graph](#).

autodiscovery

The mechanism that detects the set of **master**, **slave**, and **master/slave** servers in a **replication** configuration, and displays the **topology** in the **Server tree**.

See Also [Asset Selector](#), [master](#), [replication](#), [slave](#), [topology](#).

B

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.

See Also [Agent](#).

blackout period

A function of Event Notification, it is a time period where events do not trigger notifications via the Event Handler, in that information is not reported from a MySQL Server. Typically, this is during a maintenance period when the database might go through an unusual workload that does not require raising any **notifications**.

See Also [Agent](#), [instance](#), [notification](#).

bottleneck

An aspect or component of a system whose capacity imposes a limit on performance. In MySQL Enterprise Monitor, you identify bottlenecks in areas such as I/O or memory usage using **graphs**, and use **advisors** to automatically raise **events** when problems occur.

See Also [advisor](#), [event](#), [graph](#).

bundled

The “bundled MySQL server” refers to the MySQL server delivered with the MySQL Enterprise Monitor product, to serve as the **repository** for the collected data.

See Also [repository](#).

C

C

The C programming language.

canonical query

A synonym for **normalized query**.

See Also [normalized query](#).

chain replication

In **replication**, a kind of **topology** where each server acts as a **source/replica** server, connected to at most two other servers.

See Also [circular replication](#), [master/slave](#), [replication](#), [topology](#).

circular replication

In a **replication** configuration, a **topology** where every server is both a **source** and a **replica**, connected to two other servers, forming a ring structure. Forms a **ring** structure.

See Also [replication](#), [ring replication](#), [topology](#).

credentials

The MySQL Enterprise Monitor product interfaces with many different databases and other kinds of servers. Each of these components can have its own login and security credentials. MySQL Enterprise Monitor pulls performance data from the MySQL servers that you monitor, stores the resulting data in a **repository** that is also a MySQL server, and sends alerts by communicating through other kinds of servers such as **SMTP** for e-mail alerts and **NMS** for **SNMP** traps. It pulls support-related data from the **My Oracle Support** site. You view the results in the **UI**, which is protected by its own login and optionally by **LDAP** authentication.

See Also [LDAP](#), [NMS](#), [repository](#), [SMTP](#), [SNMP](#).

Critical

A high severity level for **alarms**. Within a **rule**, you can designate the **threshold** values that specify whether an alarm triggers a Critical, **Warning**, or **notice** alert.

See Also [advisor](#), [event](#), [Notice](#), [threshold](#), [Warning](#).

custom

User-created **rules** and **graphs**. Contrast with **standard**.

See Also [graph](#), [standard](#).

D

data collection item

Data values derived from server status variables, operating system status information, and MySQL table information. You can reference these items using mnemonic names in **expressions** when you create or edit **rules**.

See Also [expression](#).

diagnostic report

The diagnostic 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.

If you intend to communicate with MySQL Enterprise Monitor support, it is strongly recommended you provide this report with your support request.

See Also [My Oracle Support](#), [service request](#), [support issue](#).

E

event

The highest severity level for an advisor. Within an **Advisor**, you can designate the **threshold** values that specify whether an event triggers an Emergency, **Critical**, **Warning**, or **notice** event.

See Also [advisor](#), [Critical](#), [event](#), [Notice](#), [Warning](#).

event

Events are displayed if an Advisor Threshold is crossed, and are used to inform you of errors or potential problems with your implementation. Monitor events using the **Events** page, and define how events are handled by configuring **Event Handling**.

See Also [Critical](#), [Events](#), [Notice](#), [notification](#), [Warning](#).

event handler

Event handlers are conditions associated with notifications. If the condition is met, the notification is sent.

See Also [Critical](#), [event](#), [Events](#), [Notice](#), [notification](#), [Warning](#).

Events

A page within the **UI**, for monitoring **events**. See [Section 26.1, “Events”](#).

See Also [event](#).

EXPLAIN plan

A textual report showing the internal mechanisms used by a **query**, and estimates for the “cost”, such as amount of data to process, involved in each step. Performance monitoring involves checking whether queries that are slow or frequently run could be sped up or made less resource-intensive, by examining the EXPLAIN plan to check if the MySQL optimizer has chosen the most efficient ways to process indexes, order join clauses, and so on. The MySQL Enterprise Monitor product includes a number of features for visualizing EXPLAIN plans.

See Also [query](#).

expression

The part of an **Advisor** that tests a **metric** against a **threshold** value.

See Also [advisor](#), [metric](#), [threshold](#).

F

firewall

Additional considerations apply when the server you are monitoring is separated from the **Service Manager** by a firewall.

See Also [remote](#), [Service Manager](#), [SSL](#).

full table scan

An aspect of a [SELECT query](#) that often indicates a performance or scalability issue. The query scans every row in a table, rather than using an index to look up a subset of rows. It can be a non-issue for small tables that are cached in memory. It can be unavoidable when querying large tables to prepare reports. Performance issues are most likely when the table being scanned is involved in a join operation, when the query is run frequently, or when the result set only references a small fraction of the rows in the table.

To diagnose possible issues due to full table scans, choose a time period of heavy SQL activity from one of the **graphs**, use the **Query Analyzer** to locate queries that process large numbers of rows, and examine the **explain plan** for the queries. The notation in the explain plan that indicates a full table scan is `Type=ALL`.

See Also [EXPLAIN plan](#), [graph](#), [query](#), [Query Analyzer](#).

G

graph

A visual representation of server activity, resources, or other **metric** over time. You view graphs using the **Metrics** view.

See Also [metric](#).

H

heartbeat

An implementation detail from MySQL Enterprise Monitor 2.x that is no longer present. A heartbeat signal that is sent from an Agent to the Service Manager was defined as the heartbeat, and was present in the connecting URL.

See Also [Agent](#), [Service Manager](#).

host

A computer running one or more **instances** of **MySQL server**. Typically each host is monitored by a MySQL Enterprise Monitor Agent.

See Also [instance](#), [MySQL server](#).

I

instance

A `mysqld` daemon running on a MySQL server. There might be multiple instances running on the same server machine. An instance is one of the **asset** MySQL Enterprise Monitor can monitor.

The **Instances** view manages the instances, and the **MySQL Process Discovery** advisor discovers local instances on the host that the Agent is installed on.

See Also [asset](#).

instrument

A persisted **instance** of a run-time **metric** evaluation. These may store the raw metric data, or the result of an expression or function against a metric. Instruments are generally stored for things that are displayed in the GUI, such as **graph** data.

See Also [graph](#), [metric](#).

inventory

Core information about a host or MySQL server **instance**. This data is collected by the **Agent**. The inventory includes details such as the MySQL server version number, supported storage engines and **replication** configuration. The data in the inventory helps to determine what other kinds of data can be collected from the hosts and MySQL servers.

See Also [Agent](#), [instance](#), [replication](#).

J

Java

The MySQL Enterprise Monitor product is partly built on Java technology, using the **Tomcat** servlet container for the web-based GUI. It uses a JRE on the machine that runs the **Service Manager**, its performance is affected by the Tomcat configuration parameters, the Agent is written in Java, and during troubleshooting you might use diagnostic information from the Java environment.

See Also [JDBC](#), [Query Analyzer](#), [Tomcat](#).

JDBC

The underlying database API used by many **Java** applications.

See Also [Java](#).

JET

Oracle JavaScript Extension Toolkit (JET) is a collection of open source JavaScript libraries along with a set of Oracle contributed JavaScript libraries used to build applications that consume and interact with Oracle products and services.

See Also [UI](#).

L

LDAP

An authentication mechanism that can control access to the **UI**. On Linux, Unix, and macOS systems, you might have a separate LDAP server where each user has their own **credentials**. On Windows systems, the LDAP protocol is used to connect to an Active Directory server for the same authentication purposes. Typically, you map LDAP roles to corresponding MySQL Enterprise Monitor **roles**, to enable groups of users to have basic or administrative access to the Dashboard without configuring each user individually, or giving them a new user ID and password, or requiring extra work to revoke access when they leave the organization.

See Also [credentials](#), [role](#).

log file

MySQL Enterprise Monitor manages log files for the **Tomcat**, **repository**, **agent** and **Service Manager** components, as well as a configuration report pertaining to the initial installation.

See Also [repository](#), [Service Manager](#), [Tomcat](#).

Lua

A programming language that is used for parts of the Service Manager. Although you do not need to know this language to operate MySQL Enterprise Monitor, you may need to specify options containing file paths related to Lua scripts or libraries.

See Also [Agent](#), [Query Analyzer](#), [Service Manager](#).

M

master

In a **replication** configuration, a database server that sends updates to a set of **slave** servers. It typically dedicates most of its resources to write operations, leaving user queries to the slaves. In complex **topologies**, a server can be both a master and a slave, known as a **master/slave**.

See Also [master/slave](#), [replica](#), [replication](#), [slave](#), [source](#), [topology](#).

master/slave

In **replication**, a server that acts as a **slave** to receive updates from another server, and also acts as a **master** to propagate changes to another set of slave servers. Keeps the top-level master from having to service too many slaves, and allows certain kinds of changes to be propagated to a subset of slaves. This **topology** is represented as a **tree**, with all the intermediate nodes being master/slave servers.

See Also [master](#), [replication](#), [slave](#), [topology](#), [tree](#).

metric

Any property that is measured using a numeric value. Within MySQL Enterprise Monitor, such measurements can be displayed over time as a **graph**, or an **alert** can be sent when a **threshold** value is reached. Each metric is collected from an **asset**. For example, how much time is taken by a database query, or how full is the file system on a server machine.

See Also [asset](#), [graph](#), [threshold](#).

mixed replication

In a **replication** configuration, a kind of topology consisting of different combinations of **master**, **slave**, and dual-purpose **master/slave** servers.

See Also [chain replication](#), [circular replication](#), [master](#), [master/slave](#), [replication](#), [slave](#), [topology](#), [tree](#).

monitor

To view information about the state, health, activity, and history of a resource such as a MySQL server **instance**. Monitoring can help to diagnose problems, spot worrisome trends before they turn into problems, reassure when systems are operating normally, and notify when an operator needs to take corrective action. With MySQL Enterprise Monitor, the component you interact with during monitoring is the **UI**.

See Also [instance](#).

My Oracle Support

The website for filing bugs and **service requests** with Oracle Support. (Commercial customers such as MySQL Enterprise Monitor users now use the official Oracle support channel rather than the MySQL bug database.) In MySQL Enterprise Monitor, you interact with the support site through the **What's New tab**.

See Also [service request](#), [What's New tab](#).

MySQL server

The MySQL database **instance** running on a **host** computer. More than one MySQL instance can run on the same computer, either as separate `mysqld` processes or through virtualization.

See Also [instance](#).

N

.NET

A programming framework used by many Windows applications.

NMS

Acronym for Network Management System, a type of monitoring system that is separate from the MySQL Enterprise Monitor product. If your network has this kind of monitoring capability, MySQL Enterprise Monitor can notify the NMS of network issues by translating **events** into **SNMP** traps.

See Also [event](#), [SNMP](#).

normalized query

A condensed form of the query text used to treat similar queries as if they were identical, for monitoring performance. When MySQL Enterprise Monitor normalizes queries, it disregards differences in keyword capitalization, whitespace, and most comments. It replaces literal values with placeholders and transforms multi-row insert statements and **IN** clauses, to group similar statements with different parameters when measuring how much time is consumed by a particular type of query.

See Also [canonical query](#), [query](#).

Notice

A low severity level for **events**. Within an **advisor**, you can designate the **threshold** values that specify the severity level that is triggered.

See Also [advisor](#), [Critical](#), [event](#), [threshold](#), [Warning](#).

notification

The e-mail or other form of communication triggered by an **alarm**. Notifications are defined using **Event Handlers**.

See Also [event](#).

notification group

A collection of users who receive a **notification** when advisor **alerts** occur. These users might have login credentials for the **UI**, but this is not a requirement.

See Also [event](#), [event handler](#), [notification](#).

P

PHP

A scripting language typically embedded within web pages.

Q

QRTi

QRTi stands for "Query Response Time index". It is a "quality of service" measurement for each query, and it uses the Apdex formula for that calculation. For more information about Apdex, see: [Apdex on Wikipedia](#).

See Also [Query Analyzer](#).

QuAn

An abbreviation for **Query Analyzer** used in some keyword names.

See Also [Query Analyzer](#).

query

In the MySQL Enterprise Monitor context, any SQL statement whose performance might be monitored. Includes not just SQL [SELECT](#) queries, but also DML statements such as [INSERT](#), [UPDATE](#), and [DELETE](#), and DDL statements such as [CREATE TABLE](#) and [TRUNCATE TABLE](#).

See Also [normalized query](#), [slow query log](#).

Query Analyzer

The MySQL Enterprise Monitor component that tracks data about MySQL **queries** and summarizes that data using **graphs** and tables. You interact with it (for example, filtering the displayed queries or selecting a specific time period) using the **Query Analyzer tab** in the **UI**.

See Also [graph](#), [query](#), [Query Analyzer view](#).

Query Analyzer view

A page within the **UI** that displays output and controls the options for the **Query Analyzer** component. See [Chapter 32, Query Analyzer View](#) for more information.

See Also [Query Analyzer](#).

R

remote

Additional considerations apply when the server you are monitoring is separated from the **Service Manager**, for example by a wide-area network or by a **firewall**.

See Also [firewall](#), [remote monitoring](#), [SSL](#).

remote monitoring

An Agent on one host that monitors different hosts, and MySQL instances on different hosts.

Remote monitoring has limitations, versus local monitoring. Remote monitoring does not provide Operating System level data, such as CPU, file, and network utilization information.

See Also [Agent](#), [remote](#).

replica

Synonym for slave in replication topologies.

In a **replication** configuration, a database server that receives updates from a **source** server. Typically used to service user queries, to minimize the query load on the master. In complex **topologies**, a server can be both a source and a replica, known as a **source/replica**.

See Also [master](#), [master/slave](#), [replication](#), [topology](#).

replication

A set of database features that mirrors the same data across a set of servers. Used for reliability in case of server failure, and to speed up queries by dividing the work across servers. Because replication involves so many aspects of reliability and performance, it is an important aspect to monitor and has a dedicated **Replication** view in the **UI**.

See Also [topology](#).

Replication view

A view within the **UI** that monitors aspects of **replication**. See [Chapter 28, Replication Dashboard](#).

See Also [replication](#).

repository

The database that stores the monitoring data collected by the MySQL Enterprise Monitor product. It can be a separate database **instance** that is part of the MySQL Enterprise Monitor installation (the **bundled** MySQL server), or you can use an existing database of your own. The bundled MySQL server is a level of MySQL database that is fully tested with the MySQL Enterprise Monitor product, and can be kept separate from your other databases to avoid any extra load on them. You might use an existing server that has spare capacity, fast storage devices, tuned configuration parameters, a backup system, or other conveniences that can benefit the MySQL Enterprise Monitor data storage as well.

See Also [bundled](#), [instance](#).

ring replication

See [circular replication](#).

role

A level of access privilege for the **UI**. One of [manager](#) (highest privilege), [dba](#), [read-only](#), or [agent](#) (specialized privilege for sending data). Each user account registered with the Dashboard must have one of these roles. Components such as the **Agent** and the use the [agent](#) role. To simplify the process of granting **credentials** for large numbers of users, you can map **LDAP** roles to these MySQL Enterprise Monitor roles.

See Also [Agent](#), [credentials](#), [LDAP](#).

S

schedule

Determines how often each **advisor** is evaluated.

See Also [advisor](#).

scope

A dynamic filter against an **asset** class.

See Also [asset](#).

Service Manager

The core MySQL Enterprise Monitor component that receives the monitoring data from the **Agent** and **Query Analyzer** components. It displays this information through the GUI web-interface, and manages the **Advisors**, **Events**, and **Event Handlers**, as well as access to Reporting, Graphing, and Query Analysis.

See Also [advisor](#), [Agent](#), [event](#), [Query Analyzer](#).

service request

Generic term for a bug that you have filed or a support request that you have opened, on the **My Oracle Support** site. Formerly, we referred to these as **support issues**.

See Also [My Oracle Support](#), [support issue](#).

Settings

A page in the **UI** for viewing or editing configuration settings of the **Service Manager**.

See Also [Service Manager](#).

severity

A measure of the seriousness of an **Event**. Event Handling policies can take into account the severity level in determining the appropriate course of Notification action.

See Also [event handler](#).

slave

In a **replication** configuration, a database server that receives updates from a **master** server. Typically used to service user queries, to minimize the query load on the master. In complex **topologies**, a server can be both a master and a slave, known as a **master/slave**.

See Also [master](#), [master/slave](#), [replica](#), [replication](#), [source](#), [topology](#).

slow query log

A MySQL Server facility for tracking [queries](#) that consume considerable time and resources. MySQL Enterprise Monitor provides more information about query performance than the slow query log, and does not currently use the slow query log itself.

See Also [query](#).

SMTP

The e-mail protocol for sending **alerts** as e-mail messages. To send alerts this way, you configure e-mail settings similar to those in an e-mail client.

SNMP

A protocol for sending event **notifications** (“SNMP traps”) to an **NMS**. The MySQL Enterprise Monitor product can turn notifications for selected rules into SNMP traps. In contrast to typical **alerts** that are only raised when some issue occurs, SNMP traps are broadcast for all state changes, so that corrective action can be cancelled when an issue is cleared.

See Also [NMS](#), [notification](#).

source

Synonym for master in replication topologies.

In a **replication** configuration, a database server that sends updates to a set of **replica** servers. It typically dedicates most of its resources to write operations, leaving user queries to the replicas. In complex **topologies**, a server can be both a source and a replica, known as a **source/replica**.

See Also [master/slave](#), [replica](#), [replication](#), [slave](#), [topology](#).

source/replica

In **replication**, a server that acts as a **replica** to receive updates from another server, and also acts as a **source** to propagate changes to another set of replica servers. Keeps the top-level source from having to service too

many replicas, and allows certain kinds of changes to be propagated to a subset of replicas. This **topology** is represented as a **tree**, with all the intermediate nodes being source/replica servers.
See Also [replica](#), [replication](#), [source](#), [topology](#), [tree](#).

SSL

For MySQL Enterprise Monitor, the default port used for SSL connections is 18443. If you are connecting using SSL, the built-in MySQL Enterprise Monitor certificate is self-signed and may be highlighted as “unsafe” within the browser on initial connection. To prevent problems accessing the site, add an exception for the certificate to your browser for this server.

standard

The predefined **advisors** and **graphs** supplied as part of MySQL Enterprise Monitor. Contrast with **custom**.
See Also [custom](#), [graph](#).

Status Summary

The [Status Summary](#) widget is displayed on top of every page, updates dynamically, and shows current status counters for Hosts monitored, MySQL Instances monitored, MySQL Instances with invalid connection configurations, Unmonitored MySQL Instances, and Emergency Events. The counters are live links for displaying details or resolving the issues they are reporting.

support issue

Former term for bugs and service requests. In the My Oracle Support system, now known as a **service request**, which is the preferred term.
See Also [service request](#).

T

table scan

See [full table scan](#).

threshold

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.
See Also [Critical](#), [event](#), [expression](#), [Notice](#), [Warning](#).

Tomcat

Apache Tomcat is a component of the **Service Manager**. It is the servlet engine that powers the **UI**.
See Also [Apache](#), [Java](#), [Service Manager](#).

topology

In a **replication** configuration, the way in which the different **source**, **replica**, and dual-purpose **source/replica** servers are connected. In MySQL Enterprise Monitor, the configurations are classified in the **Replication view** as one of [TREE](#), [RING](#), or [MIXED](#).
See Also [master](#), [master/slave](#), [replication](#), [Replication view](#), [slave](#).

topology view

A graphical representation of the replication topology displayed in a dedicated **Topology** view.
See Also [replica](#), [replication](#), [Replication view](#), [source](#).

tree

A data structure often used to represent relationships between MySQL servers. In the **UI**, servers are displayed in the **Server Tree** sidebar. In a **replication** configuration, setting up some machines as dual-purpose **master/slave** servers produces a nesting relationship that is represented in the Server Tree.

See Also [Asset Selector](#), [master/slave](#), [replication](#).

trending

Predicting when a problem may occur. For example, predicting that a disk is expected to run out of disk space 30 days in the future.

U

UI

The UI is a web-based interface to the MySQL Enterprise **Service Manager**. The back end is a **Java** application powered by the **Tomcat** server.

UUID

A unique identifier used to distinguish each MySQL **instance**, host machine, and **agent**. Because there is so much flexibility in spreading components across multiple systems or running multiple instances and agents on the same system, the combination of these different UUIDs identifies where information came from and the source of any issues. Always generate a new UUID for any one of these components, rather than copying or reusing an existing UUID value.

In MySQL 5.5, or earlier, the UUID for the server is stored in the table `mysql.inventory`. When a component such as the Agent connects to that MySQL server, the applicable MySQL user must have privileges to read this table.

See Also [Agent](#), [instance](#).

W

Warning

A medium severity level for **alarms**. Within a **rule**, **you can designate the threshold** values that specify for each severity level.

See Also [advisor](#), [Critical](#), [event](#), [Notice](#), [notification](#), [threshold](#).

What's New tab

A page within the **UI** that provides updates and news related to MySQL Enterprise Monitor and your **My Oracle Support** account.

See Also [My Oracle Support](#).

Index

A

- access control, 343
- Administration Advisors, 157
 - 32-Bit Binary Running on 64-Bit AMD Or Intel System, 157
 - Binary Log Debug Information Disabled, 158
 - Binary Logging Is Limited, 158
 - Binary Logging Not Enabled, 158
 - Binary Logging Not Synchronized To Disk At Each Write, 158
 - Binary Logs Automatically Removed Too Quickly, 159
 - Database May Not Be Portable Due To Identifier Case Sensitivity, 159
 - Event Scheduler Disabled, 159
 - General Query Log Enabled, 159
 - Host Cache Size Not Sufficient, 160
 - In-Memory Temporary Table Size Limited By Maximum Heap Table Size, 160
 - InnoDB Status Truncation Detected, 160
 - InnoDB Strict Mode Is Off, 161
 - InnoDB Tablespace Cannot Automatically Expand, 161
 - InnoDB Transaction Logs Not Sized Correctly, 161
 - Multiple Threads Used When Repairing MyISAM Tables, 161
 - MySQL Server No Longer Eligible For Oracle Premier Support, 161
 - Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 162
 - No Value Set For MyISAM Recover Options, 162
 - Table Cache Set Too Low For Startup, 162
 - Time Zone Data Not Loaded, 162
 - Warnings Not Being Logged, 163
- adminpassword option, 73
- adminuser option, 72
- advisor, 343
- advisor category, 343
- Advisors, 191
 - 32-Bit Binary Running on 64-Bit AMD Or Intel System, 157
 - Account Has An Overly Broad Host Specifier, 205
 - Account Has Global Privileges, 205
 - Account Has Old Insecure Password Hash, 185
 - Account Requires Unavailable Authentication Plug-ins, 185
 - Agent Health Advisor, 191
 - Attempted Connections To The Server Have Failed, 164
 - AUTO_INCREMENT Field Limit Nearly Reached, 180
 - Average Statement Execution Time Advisor, 200
 - Binary Log Checksums Disabled, 176
 - Binary Log Debug Information Disabled, 158
 - Binary Log File Count Exceeds Specified Limit, 176
 - Binary Log Row Based Images Excessive, 176
 - Binary Log Space Exceeds Specified Limit, 176
 - Binary Log Usage Exceeding Disk Cache Memory Limits, 170
 - Binary Logging Is Limited, 158
 - Binary Logging Not Enabled, 158
 - Binary Logging Not Synchronized To Disk At Each Write, 158
 - Binary Logs Automatically Removed Too Quickly, 159
 - Cluster Data Node Redo Buffer Space Getting Low, 165
 - Cluster Data Node Redo Log Space Getting Low, 165
 - Cluster Data Node Undo Buffer Space Getting Low, 165
 - Cluster Data Node Undo Log Space Getting Low, 166
 - Cluster DiskPageBuffer Hit Ratio Is Low, 166
 - CPU Utilization Advisor, 196
 - Database May Not Be Portable Due To Identifier Case Sensitivity, 159
 - Database-level Privileges on All Databases, 205
 - Duplicate Host Identity, 196
 - Duplicate MySQL Server UUID, 196
 - Event Scheduler Disabled, 159
 - Excessive Disk Temporary Table Usage Detected, 170
 - Excessive Number of Locked Processes, 171
 - Excessive Number of Long Running Processes, 171
 - Excessive Number of Long Running Processes Locked, 171
 - Excessive Percentage Of Attempted Connections To The Server Have Failed, 164
 - Filesystem Free Space Advisor, 198
 - Flush Time Set To Non-Zero Value, 171
 - General Query Log Enabled, 159
 - Group Replication, 202
 - Group Replication Configuration, 202
 - Group Replication Status, 203
 - Host Cache Size Not Sufficient, 160
 - HTTP Server Performance, 168
 - In-Memory Temporary Table Size Limited By Maximum Heap Table Size, 160
 - Indexes Not Being Used Efficiently, 171
 - InnoDB Buffer Cache Has Sub-Optimal Hit Rate, 166
 - InnoDB Buffer Pool Writes May Be Performance Bottleneck, 172
 - InnoDB Flush Method May Not Be Optimal, 172
 - InnoDB Log Buffer Flushed To Disk After Each Transaction, 172
 - InnoDB Log Waits May Be Performance Bottleneck, 173
 - InnoDB Not Using Newest File Format, 172
 - InnoDB Status Truncation Detected, 160
 - InnoDB Strict Mode Is Off, 161

InnoDB Tablespace Cannot Automatically Expand, 161

InnoDB Transaction Logs Not Sized Correctly, 161

Insecure Password Authentication Option Is Enabled, 185

Insecure Password Generation Option Is Enabled, 186

Key Buffer Size May Not Be Optimal For Key Cache, 166

LOCAL Option Of LOAD DATA Statement Is Enabled, 186

Master Not Verifying Checksums When Reading From Binary Log, 177

Maximum Connection Limit Nearing Or Reached, 164

Multiple Threads Used When Repairing MyISAM Tables, 161

MyISAM Concurrent Insert Setting May Not Be Optimal, 173

MyISAM Indexes Found with No Statistics, 182

MySQL Agent Memory Usage Excessive, 163

MySQL Agent Not Reachable, 163

MySQL Availability, 164

MySQL Enterprise Backup Health Advisor, 194

MySQL Process Discovery Advisor, 194

MySQL Server Has Been Restarted, 165

MySQL Server No Longer Eligible For Oracle Premier Support, 161

MySQL User Account, 205

Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 162

No Value Set For MyISAM Recover Options, 162

Object Changed: Database Has Been Altered, 181

Object Changed: Database Has Been Created, 181

Object Changed: Database Has Been Dropped, 181

Object Changed: Function Has Been Created, 181

Object Changed: Function Has Been Dropped, 182

Object Changed: Index Has Been Created, 182

Object Changed: Index Has Been Dropped, 182

Object Changed: Table Has Been Altered, 183

Object Changed: Table Has Been Created, 183

Object Changed: Table Has Been Dropped, 183

Object Changed: User Has Been Dropped, 184

Object Changes Detected, 182

Policy-Based Password Validation Does Not Perform Dictionary Checks, 186

Policy-Based Password Validation Is Weak, 187

Policy-Based Password Validation Not Enabled, 187

Prepared Statements Not Being Closed, 173

Prepared Statements Not Being Used Effectively, 173

Privilege Alterations Detected: Privileges Granted, 187

Privilege Alterations Detected: Privileges Revoked, 187

Privilege Alterations Have Been Detected, 187

Query Analysis Reporting, 201

Query Cache Has Sub-Optimal Hit Rate, 167

Query Cache Is Excessively Fragmented, 174

Query Cache Potentially Undersized, 167

Query Pileup Advisor, 201, 201

Replica Execution Position Too Far Behind, 178

Replication Configuration Advisor, 177

Replication Status Advisor, 177

Replication Too Far Behind, 178

Root Account Can Login Remotely, 188

Root Account Without Password, 188

Server Contains Default "test" Database, 188

Server Has Accounts Without A Password, 188

Server Has Anonymous Accounts Not Being Used As Proxies, 189

Server Has No Locally Authenticated Root User, 189

Server Includes A Root User Account, 189

Server-Enforced Data Integrity Checking Disabled, 182

Server-Enforced Data Integrity Checking Not Strict, 183

SHA-256 Password Authentication Not Enabled, 188

Slave Detection Of Network Outages Too High, 177

Slave Has Login Accounts With Inappropriate Privileges, 178

Slave Not Configured As Read Only, 178

Slave Not Verifying Checksums When Reading From Relay Log, 178

Slave Relay Log Space Is Very Large, 179, 179

Slave SQL Processing Not Multi-Threaded, 179

Slave Without REPLICATION SLAVE Accounts, 180

SQL Statement Generates Warnings or Errors, 201

Support Diagnostics, 169

Symlinks Are Enabled, 189

sys Schema Install Advisor, 196

Table Cache Not Optimal, 167

Table Cache Set Too Low For Startup, 162

Table Lock Contention Excessive, 174

Tables Found with No Primary or Unique Keys, 184

Thread Cache Not Enabled, 174

Thread Cache Size May Not Be Optimal, 167

Thread Pool Stall Limit Too Low, 174

Thread Pooling Not Enabled, 175

Time Zone Data Not Loaded, 162

Too Many Concurrent Queries Running, 175

Unsupported Agent Version, 169

User Has Rights To Database That Does Not Exist, 189

User Has Rights To Table That Does Not Exist, 190

Users Can View All Databases On MySQL Server, 190

Warnings Not Being Logged, 163

advisors
 creating, 285, 290

agent
 configuration utilities, 89
 installation, 41

- troubleshooting, 52
- Agent, 343
 - Managing agents, 131
- Agent Advisors, 163
 - Agent Health Advisor, 191
 - MySQL Agent Memory Usage Excessive, 163
 - MySQL Agent Not Reachable, 163
- agent_autocreate option, 79
- agent_installtype option, 80
- agentpassword option, 75
- agentservicename option, 81
- agentuser option, 76
- Apache, 343
- Apple OS X, 49, 94, 327
- architecture, 7
- asset, 343
- Asset management
 - inventory, 131
- Asset Selector, 343
- Assets
 - Deleting Assets, 135
 - management, 131
- autodiscovery, 343
- Availability Advisors, 163
 - Attempted Connections To The Server Have Failed, 164
 - Excessive Percentage Of Attempted Connections To The Server Have Failed, 164
 - Maximum Connection Limit Nearing Or Reached, 164
 - MySQL Availability, 164
 - MySQL Server Has Been Restarted, 165

B

- backlog, 344
- backup
 - restore, 57
- Backup Advisors
 - MySQL Enterprise Backup Health Advisor, 194
- Backup Dashboard
 - Backup Details, 255
 - filter, 255
 - Group Overview, 253
 - History, 254
 - Instance Overview, 253
- Backup Details
 - Backup Command, 257
 - Backup Configuration, 256
 - Backup Status, 256
 - InnoDB Configuration, 257
 - Progress Log, 257
- backupdir option, 69
- backups, 63
- blackout

- blackout period, 296
- blackout scripting, 298
- command line, 298
- curl, 298
- blackout period, 344
- bottleneck, 344
- browsers, 31
- bundled, 344

C

- C, 344
- canonical query, 344
- chain replication, 344
- checkmysqlhost option, 76
- circular replication, 344
- Cluster
 - Managing clusters, 134
- Cluster Advisors
 - Cluster Data Node Redo Buffer Space Getting Low, 165
 - Cluster Data Node Redo Log Space Getting Low, 165
 - Cluster Data Node Undo Buffer Space Getting Low, 165
 - Cluster Data Node Undo Log Space Getting Low, 166
 - Cluster DiskPageBuffer Hit Ratio Is Low, 166
- config.properties file, 328
- Configuration, 101
- configuration files, 327
- configuration utilities
 - agent, 89
 - agent.bat, 89
 - agent.sh, 89
 - config.bat, 87
 - config.sh, 87
 - service manager, 87
- Configuration View
 - Configuration Filters, 109
- createBackup option, 76
- createDataBackup option, 69
- credentials, 344
- Critical, 344
- custom, 345
- custom data collection
 - customizing advisors, 292

D

- data collection item, 345
- dbhost option, 74
- dbname option, 74
- dbpool.default.initialSize option, 330
- dbpool.default.maxActive option, 330
- dbpool.default.maxIdle option, 330
- dbpool.default.maxWaitMillis option, 330

dbpool.default.minEvictableIdleTimeMillis option, 331
dbpool.default.minIdle option, 330
dbpool.default.timeBetweenEvictionRunsMillis option, 330
dbpool.ui.initialSize option, 329
dbpool.ui.maxActive option, 329
dbpool.ui.maxIdle option, 329
dbpool.ui.maxWaitMillis option, 329
dbpool.ui.minEvictableIdleTimeMillis option, 330
dbpool.ui.minIdle option, 329
dbpool.ui.timeBetweenEvictionRunsMillis option, 330
--dbport option, 73
--debuglevel option, 70, 76
--debugtrace option, 70, 77
Deleting Assets, 135
diagnostic report, 341, 345
Diagnostics Report, 341

E

event, 345, 345
event handler, 345
Events, 106, 345
EXPLAIN plan, 345
expression, 345
expressions, 288

F

FAQs, 335
Filtering
 Named Filters, 108
firewall, 346
firewall issues, 52
--forceRestart option, 74
full table scan, 346

G

--generalpassword option, 80
--generaluser option, 79
graph, 346
graphs
 creating, 285, 291
Group Replication Advisors
 Group Replication Configuration, 202
 Group Replication Status, 203

H

heartbeat, 346
--help option, 69, 81
host, 346
Hosts
 Managing hosts, 132

I

InnoDB Buffer Pool Usage Report, 282
installation
 agent, 41
 backup, 57
 post-install tasks, 59
 service manager, 33
 unattended, 67
 uninstalling, 93
--installdir option, 71, 77
--installer option, 71
--installer-language option, 77
instance, 346
Instance Status
 Status Summary, 108
instrument, 346
Inventory
 Deleting Assets, 135
inventory, 347
Inventory management
 assets, 131

J

Java, 347
Java VM
 installation, 33
JDBC, 347
JET, 347

L

LDAP, 347
--limitedpassword option, 80
--limiteduser option, 80
Linux, 50, 327
locale, 104
log file, 347
 MySQL Enterprise Service Manager, 327
Logout
 User Menu, 107
Lua, 347

M

Mac OS X, 49, 94, 327
macOS, 49, 94, 327
--managerhost option, 77
--managerport option, 77
master, 347
master/slave, 348
Memory Usage Advisors, 166
 InnoDB Buffer Cache Has Sub-Optimal Hit Rate, 166
 Key Buffer Size May Not Be Optimal For Key Cache, 166
 Query Cache Has Sub-Optimal Hit Rate, 167

- Query Cache Potentially Undersized, 167
- Table Cache Not Optimal, 167
- Thread Cache Size May Not Be Optimal, 167
- Memory Usage Report, 281
- Menus
 - Metrics, 106
- metric, 348
- MIB file, 327
- mixed replication, 348
- mode option, 70, 78
- monitor, 348
 - installation, 33
- Monitoring and Support Advisors
 - sys Schema Install Advisor, 196
- Monitoring and Support Services Advisors, 168
 - Duplicate Host Identity, 196
 - Duplicate MySQL Server UUID, 196
 - HTTP Server Performance, 168
 - MySQL Process Discovery Advisor, 194
 - Support Diagnostics, 168
 - Unsupported Agent Version, 169
- My Oracle Support, 348
- MySQL Enterprise Backup, 63
- MySQL Server
 - installation, 33
- MySQL server, 348
- mysql-monitor-agent.log file, 331
- mysqlconnectiongroup option, 81
- mysqlconnmethod option, 78
- mysqldump, 63
- mysqlhost option, 79
- mysql-identity-source option, 72
- mysqlpassword option, 79
- mysqlport option, 79
- mysqlsocket option, 79
- mysqluser option, 79
- mysql_installation_type option, 73

N

- Named Filters
 - Creating Filters, 108
- NDB Cluster Advisors, 165
- NDB Cluster Memory Usage Report
 - NDB Cluster, 283
- .NET, 348
- NMS, 348
- normalized query, 348
- Notice, 349
- notification, 349
- notification group, 349

O

- Operating System Advisors, 169

- CPU Utilization Advisor, 196
- Filesystem Free Space Advisor, 198
- optionfile option, 69, 80
- OS X, 49, 94, 327
- overview, 7
- Overview, 243

P

- Performance Advisors, 169
 - Binary Log Usage Exceeding Disk Cache Memory Limits, 170
 - Excessive Disk Temporary Table Usage Detected, 170
 - Excessive Number of Locked Processes, 171
 - Excessive Number of Long Running Processes, 171
 - Excessive Number of Long Running Processes Locked, 171
 - Flush Time Set To Non-Zero Value, 171
 - Indexes Not Being Used Efficiently, 171
 - InnoDB Buffer Pool Writes May Be Performance Bottleneck, 172
 - InnoDB Flush Method May Not Be Optimal, 172
 - InnoDB Log Buffer Flushed To Disk After Each Transaction, 172
 - InnoDB Log Waits May Be Performance Bottleneck, 172
 - InnoDB Not Using Newest File Format, 172
 - MyISAM Concurrent Insert Setting May Not Be Optimal, 173
 - Prepared Statements Not Being Closed, 173
 - Prepared Statements Not Being Used Effectively, 173
 - Query Cache Is Excessively Fragmented, 174
 - Table Lock Contention Excessive, 174
 - Thread Cache Not Enabled, 174
 - Thread Pool Stall Limit Too Low, 174
 - Thread Pooling Not Enabled, 175
 - Too Many Concurrent Queries Running, 175
- performance schema, 305
- performance tuning, 83
- PHP, 349
- ports
 - MySQL server for monitoring, 79
 - MySQL server for repository, 73
 - service manager, 71, 77
 - Tomcat, 71
 - Tomcat SSL, 72
- post-install tasks, 59

Q

- QRTi, 349
- QuAn, 349
- query, 349
- Query Analysis Advisors, 200
 - Average Statement Execution Time Advisor, 200

- Query Analysis Reporting, 201
- Query Pileup Advisor, 201, 201
- SQL Statement Generates Warnings or Errors, 201
- Query Analyzer, 106, 305, 349
 - Query Details, 311
 - supplying query data, 305
- Query Analyzer view, 349

R

- remote, 349
- remote monitoring, 350
- replica, 350
- Replication, 259
- replication, 350
- Replication Advisors, 175
 - Binary Log Checksums Disabled, 176
 - Binary Log File Count Exceeds Specified Limit, 176
 - Binary Log Row Based Images Excessive, 176
 - Binary Log Space Exceeds Specified Limit, 176
 - Master Not Verifying Checksums When Reading From Binary Log, 177
 - Replica Execution Position Too Far Behind, 178
 - Replication Configuration Advisor, 177
 - Replication Status Advisor, 177
 - Replication Too Far Behind, 178
 - Slave Detection Of Network Outages Too High, 177
 - Slave Has Login Accounts With Inappropriate Privileges, 178
 - Slave Not Configured As Read Only, 178
 - Slave Not Verifying Checksums When Reading From Relay Log, 178
 - Slave Relay Log Space Is Very Large, 179, 179
 - Slave SQL Processing Not Multi-Threaded, 179
 - Slave Without REPLICATION SLAVE Accounts, 180
- Replication view, 350
- repository, 350
 - database name, 74
- restartImmediately option, 76
- restore
 - backup, 57
- restoring
 - backup, 57
- role, 350
- rules
 - blackout periods, 296
 - creating, 285
 - variable substitution, 288

S

- schedule, 350
- Schema Advisors, 180
 - AUTO_INCREMENT Field Limit Nearly Reached, 180
 - MyISAM Indexes Found with No Statistics, 182

- Object Changed: Database Has Been Altered, 181
- Object Changed: Database Has Been Created, 181
- Object Changed: Database Has Been Dropped, 181
- Object Changed: Function Has Been Created, 181
- Object Changed: Function Has Been Dropped, 181
- Object Changed: Index Has Been Created, 182
- Object Changed: Index Has Been Dropped, 182
- Object Changed: Table Has Been Altered, 183
- Object Changed: Table Has Been Created, 183
- Object Changed: Table Has Been Dropped, 183
- Object Changed: User Has Been Dropped, 184
- Object Changes Detected, 182
- Server-Enforced Data Integrity Checking Disabled, 182
- Server-Enforced Data Integrity Checking Not Strict, 183
- Tables Found with No Primary or Unique Keys, 184
- scope, 350
- Security Advisors, 184, 204
 - Account Has Old Insecure Password Hash, 185
 - Account Requires Unavailable Authentication Plug-ins, 185
 - Insecure Password Authentication Option Is Enabled, 185
 - Insecure Password Generation Option Is Enabled, 186
 - LOCAL Option Of LOAD DATA Statement Is Enabled, 186
 - MySQL Enterprise Audit Plugin, 204
 - MySQL User Account, 205
 - Policy-Based Password Validation Does Not Perform Dictionary Checks, 186
 - Policy-Based Password Validation Is Weak, 186
 - Policy-Based Password Validation Not Enabled, 187
 - Privilege Alterations Detected: Privileges Granted, 187
 - Privilege Alterations Detected: Privileges Revoked, 187
 - Privilege Alterations Have Been Detected, 187
 - Root Account Can Login Remotely, 188
 - Root Account Without Password, 188
 - Server Contains Default "test" Database, 188
 - Server Has Accounts Without A Password, 188
 - Server Has Anonymous Accounts Not Being Used As Proxies, 188
 - Server Has No Locally Authenticated Root User, 189
 - Server Includes A Root User Account, 189
 - SHA-256 Password Authentication Not Enabled, 188
 - Symlinks Are Enabled, 189
 - User Has Rights To Database That Does Not Exist, 189
 - User Has Rights To Table That Does Not Exist, 190
 - Users Can View All Databases On MySQL Server, 190
- service manager
 - configuration utility, 87
 - database name, 74

- installation, 33
- Service Manager, 351
- service request, 351
- services
 - starting and stopping, 37
- Settings, 351
- severity, 351
- slave, 351
- slow query log, 351
- SMTP, 351
- SNMP, 351
- SNMP traps, 327
- source, 351
- source/replica, 351
- sql_mode, 51
- SSH tunneling, 52
- SSL, 72, 352
- standard, 352
- starting
 - MySQL Enterprise Monitor service, 38
 - MySQL Enterprise Monitor services, 37
- Status Summary, 352
 - Instance Status, 108
- stopping
 - MySQL Enterprise Monitor service, 38
 - MySQL Enterprise Monitor services, 37
- support files
 - diagnostic report, 341
- support issue, 352
- system-size option, 71

T

- Table Statistics Report, 278
- threshold, 352
- thresholds, 288
- timezone, 104
- Tomcat, 352
 - installation, 33
 - starting and stopping, 38
- tomcatport option, 71
- tomcatsslport option, 72
- topology, 352
- topology view, 352
- tree, 352
- trending, 353
- troubleshooting, 52
- tuning, 83

U

- UI, 353
- unattended installation, 67
- unattendedmodeui option, 73, 80
- uninstalling, 93, 93

- Unix, 50, 327
- User Menu
 - Logout, 107
 - User Preferences, 107
- User Preferences
 - User Menu, 107
- User Statistics Report, 280
- UUID, 353

V

- variable substitution, 288
- version option, 69, 80

W

- Warning, 353
- Web browsers, 31
- What's New tab, 353
- Wiki markup, 289
- Windows, 38, 48, 93, 327

