Abstract

This manual documents MySQL Enterprise Monitor version 3.4.10.

For notes detailing the changes in each release, see the MySQL Enterprise Monitor 3.4 Release Notes.

For legal information, including licensing information, see the Preface and Legal Notices.

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

Document generated on: 2019-04-16 (revision: 7756)
Table of Contents

Preface and Legal Notices ........................................................................................................ xiii
I Getting Started with MySQL Enterprise Monitor ................................................................. 1
  1 MySQL Enterprise Monitor Introduction and Architecture .................................................. 5
     1.1 MySQL Enterprise Monitor Component Overview ......................................................... 5
     1.2 MySQL Enterprise Monitor Agent .................................................................................. 6
     1.3 MySQL Enterprise Service Manager .............................................................................. 7
     1.4 MySQL Enterprise Monitor Proxy and Aggregator ........................................................ 10
     1.5 Data Collection .............................................................................................................. 11
     1.6 Performance Monitoring ................................................................................................ 12
     1.7 Group Replication Monitoring ....................................................................................... 12
     1.8 Performance Monitoring for Group Replication .............................................................. 12
     1.9 Limitations and Scenarios ............................................................................................... 12
     1.10 Conclusion .................................................................................................................. 13
  2 What’s New in 3.4 ............................................................................................................... 14
     2.1 Performance .................................................................................................................. 15
     2.2 Group Replication ........................................................................................................... 15
     2.3 Table Statistics Report ................................................................................................. 15
II Installing MySQL Enterprise Monitor .................................................................................... 17
     3 Installation Prerequisites ................................................................................................. 18
     3.1 Installer Files ................................................................................................................... 18
     3.2 Prerequisites .................................................................................................................. 19
     3.2.1 System Requirements ............................................................................................... 20
     3.2.2 Supported Platforms ............................................................................................... 20
     3.2.3 MySQL Enterprise Monitor Repository .................................................................. 21
     3.3 Credentials Required for Installation ............................................................................ 22
     3.3.1 Existing Users ........................................................................................................... 22
     3.3.2 Users Created During Installation ........................................................................... 22
     3.3.3 Users Created on First Log-in .................................................................................. 22
     3.4 Supported Browsers ...................................................................................................... 23
     4 Service Manager Installation ............................................................................................ 24
     4.1 MySQL Enterprise Monitor Installation Types ............................................................... 25
     4.2 MySQL Enterprise Service Manager Graphical Installation Wizard ............................... 26
     4.3 Text-Based Installation .................................................................................................. 29
     4.4 Starting/Stopping the MySQL Enterprise Monitor Services ....................................... 30
     4.5 MySQL Enterprise Service Manager Configuration Settings ...................................... 31
     5 Monitor Agent Installation ............................................................................................... 32
     5.1 General Agent Related Notes ....................................................................................... 33
     5.2 Creating MySQL User Accounts for the Monitor Agent ............................................... 34
     5.3 Java Considerations on Linux ....................................................................................... 36
     5.4 Monitoring Agent Graphical Installation Wizard ............................................................ 38
     5.5 Starting/Stopping MySQL Enterprise Monitor Agent .................................................. 40
     5.5.1 Starting/Stopping the Agent on Windows ................................................................. 40
     5.5.2 Starting/Stopping the Agent on Mac OS X ............................................................... 41
     5.5.3 Starting/Stopping the Agent on Unix ....................................................................... 42
     5.5.4 sql_mode .................................................................................................................. 43
     5.6 Monitoring Multiple MySQL Servers ............................................................................. 45
     5.7 Configuring an Agent to Monitor a Remote MySQL Server ........................................... 45
     5.8 Monitoring Outside the Firewall with an SSH Tunnel ...................................................... 46
     5.9 HTTP Connection Timeout ........................................................................................... 46
     5.10 Troubleshooting the Agent ........................................................................................... 46
     5.11 Agent Backlog ............................................................................................................... 47
     6 Upgrading MySQL Enterprise Monitor Installations .......................................................... 49
     6.1 General considerations when upgrading MySQL Enterprise Monitor .......................... 49
     6.2 Upgrading to MySQL Enterprise Monitor 3.4 ............................................................... 50
     6.3 Restoring from Backup ................................................................................................... 51
     7 Post-installation Considerations ....................................................................................... 53
     7.1 General Considerations ................................................................................................. 53
# List of Figures

1.1 MySQL Enterprise Monitor Architecture ................................................................. 5
1.2 MySQL Enterprise Monitor Agentless Architecture .................................................. 6
11.1 MySQL Enterprise Monitor Proxy and Aggregator Architecture ................................ 89
12.1 Plug-in for PHP and Aggregator Architecture ....................................................... 101
12.2 Connector Plug-in Architecture ............................................................................. 106
12.3 Connector Plug-in Architecture ............................................................................. 110
14.1 Welcome to MySQL Enterprise Monitor page ........................................................... 125
14.2 Status Summary ...................................................................................................... 129
15.1 Overview Dashboard .............................................................................................. 135
15.2 Group Overview Filter Configuration ..................................................................... 137
16.1 Add Instance Connection Settings ......................................................................... 144
16.2 Add Instance Encryption Settings ......................................................................... 145
16.3 Add Instance Advanced Settings .......................................................................... 146
16.4 MySQL Instance Filter .......................................................................................... 148
17.1 Group Management Page ....................................................................................... 151
18.1 Backup Details ...................................................................................................... 156
19.2 Replication Topology Graph .................................................................................. 161
19.3 Group Replication Single Group Topology .............................................................. 162
19.4 Group Replication Multiple Group Topology .......................................................... 163
19.5 Replication Topology Statistics ............................................................................. 167
20.1 Optimizing Graph Data Status .............................................................................. 171
20.2 Database File I/O By File ....................................................................................... 173
20.3 Database File I/O By Wait Type Report ................................................................... 174
20.4 Database File I/O By Wait Type Graphs .................................................................. 174
20.5 Database File I/O By Thread ................................................................................. 175
20.6 Processes Report .................................................................................................... 176
20.7 Table Statistics Report - Table View ..................................................................... 177
20.8 Table Statistics Report - Treemap View ................................................................. 177
21.1 Advisors Page ........................................................................................................ 179
21.2 Advisor Menu Control .......................................................................................... 181
21.3 Advisor Pop-up Menu ........................................................................................... 181
21.4 Advisor Pop-up Menu ........................................................................................... 183
21.5 Agent Health - General ......................................................................................... 184
21.6 Threshold Definitions Example ............................................................................ 185
22.1 Events Page with Filter ......................................................................................... 188
22.2 Event Handlers section .......................................................................................... 192
22.3 Email Notification Groups section ....................................................................... 193
22.4 Create Group Dialog ............................................................................................ 194
22.5 Email Settings section ........................................................................................... 195
22.6 SNMP Settings section .......................................................................................... 196
24.1 Agent Health - General ......................................................................................... 236
24.2 Agent Health - General ......................................................................................... 237
24.3 Agent Health - Backlog .......................................................................................... 238
24.4 CPU Usage ............................................................................................................ 241
24.5 CPU Outliers ......................................................................................................... 242
24.6 Filesystem - General .............................................................................................. 242
24.7 Filesystem - Estimated Full Capacity .................................................................... 243
24.8 Filesystem - Percentage of Space ......................................................................... 243
24.9 Filesystem - Percentage Used in Time Range .......................................................... 244
24.10 Group Replication Configuration Advisor ............................................................ 246
24.11 Group Replication Status Advisor ................................................................. 247
25.1 Core Monitored Assets ...................................................................................... 253
25.2 Monitoring Services Permissions ..................................................................... 255
26.1 Strict Permission Set Overview ....................................................................... 264
26.2 Strict Permission Set Grouped ........................................................................ 268
27.1 Server Hostname .............................................................................................. 269
27.2 Customize MySQL Server Name ..................................................................... 270
27.3 Data Purge Behavior ......................................................................................... 271
27.4 My Oracle Support Credentials ...................................................................... 272
27.5 HTTP Proxy Settings ......................................................................................... 272
27.6 External Authentication Settings: LDAP ......................................................... 274
29.1 MySQL Enterprise Monitor User Interface: Query Analyzer ........................ 300
### List of Tables

3.1 Required Disk Space ................................................................. 18
4.1 Installation Parameters .......................................................... 26
7.1 Agent SSL Configuration Options ................................................ 56
8.1 MySQL Enterprise Service Manager Installer Options ....................... 62
8.2 MySQL Enterprise Monitor Agent Installer Options .......................... 69
9.1 Apache Tomcat Environment File Location (default) ......................... 79
9.2 Installation Parameters ........................................................... 79
9.3 MySQL Enterprise Service Manager repository configuration file location (default) .......................................................... 80
9.4 Configuration utility location (default) ........................................ 81
9.5 Apache Tomcat configuration file location (default) ......................... 81
10.1 Default Location of Agent Configuration Utility .............................. 83
10.2 Service Manager Configuration Utilities ...................................... 84
10.3 Service Manager Certificate Utilities ........................................ 85
10.4 Default Location of Agent Configuration Utility .............................. 86
10.5 Agent Connection Utility ........................................................ 86
10.6 MySQL Enterprise Monitor Agent Configuration Utility .................... 88
11.1 MySQL Enterprise Monitor Proxy and Aggregator Installer Options ....... 94
11.2 Proxy and Aggregator Help Options ........................................... 98
11.3 Proxy and Aggregator Command Line Options ................................ 98
11.4 aggr-module Options ........................................................... 99
11.5 proxy-module Options .......................................................... 99
12.1 Connector/PHP Properties ...................................................... 104
12.2 MySQL Plug-in for Connector/J Properties .................................. 107
12.3 MySQL Plug-in for Connector/J SSL Properties .............................. 108
13.1 MySQL Enterprise Monitor Uninstaller Options .............................. 117
15.1 Group Overview Filter Configuration ........................................... 137
16.1 Bad Connection List ............................................................ 140
16.2 Unreachable Agents ............................................................ 140
16.3 Unmonitored MySQL Instances ............................................... 141
16.4 MySQL Instance Details ........................................................ 142
16.5 Connection Settings Tab ....................................................... 144
16.6 Encryption Settings Tab ........................................................ 146
16.7 Advanced Settings .............................................................. 146
16.8 MySQL Instance Filter ............................................................ 148
19.1 Group Status Columns ........................................................... 163
20.1 Timeseries Graph Filter .......................................................... 170
21.1 Advisor Page Controls ........................................................... 180
21.2 Advisor Information Listing ..................................................... 181
21.3 Advisor Edit Menu Controls .................................................... 182
21.4 Advisor Filter Controls .......................................................... 183
22.1 Events Filter Controls ............................................................ 188
22.2 Events List Columns .............................................................. 189
22.3 Event State Icons .............................................................. 189
22.4 Event Handlers Controls ........................................................ 192
22.5 Email Notification Groups List Controls ...................................... 193
22.6 Create Group Controls ........................................................... 194
22.7 Email Settings Controls ........................................................ 195
22.8 SNMP Settings Controls ........................................................ 196
22.9 Create Event Handler Controls ................................................ 197
24.1 MySQL Process Discovery Advisor Controls .................................. 239
26.1 Manager Role Definition ........................................................ 262
Preface and Legal Notices

This manual documents the MySQL Enterprise Monitor version 3.4.10.

Licensing information—MySQL Enterprise Monitor. This product may include third-party software, used under license. See MySQL Enterprise Monitor 3.4 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. See MySQL Enterprise Monitor Proxy and Aggregator 3.4 License Information User Manual for licensing information, including licensing information relating to third-party software that may be included in this release of Proxy and Aggregator.

Legal Notices

Copyright © 2005, 2019, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

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

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
Part I Getting Started with MySQL Enterprise Monitor
# Table of Contents

1 MySQL Enterprise Monitor Introduction and Architecture ................................................................. 5  
  1.1 MySQL Enterprise Monitor Component Overview ............................................................................ 5  
  1.2 MySQL Enterprise Monitor Agent ................................................................................................ 6  
  1.3 MySQL Enterprise Service Manager ............................................................................................. 7  
  1.4 MySQL Enterprise Monitor Proxy and Aggregator ......................................................................... 10  
2 What's New in 3.4 ............................................................................................................................... 11  
  2.1 Performance .................................................................................................................................... 11  
  2.2 Group Replication .......................................................................................................................... 11  
  2.3 Table Statistics Report ..................................................................................................................... 11
Chapter 1 MySQL Enterprise Monitor Introduction and Architecture

Table of Contents

1.1 MySQL Enterprise Monitor Component Overview .................................................. 5
1.2 MySQL Enterprise Monitor Agent ........................................................................ 6
1.3 MySQL Enterprise Service Manager ..................................................................... 7
1.4 MySQL Enterprise Monitor Proxy and Aggregator .................................................. 10

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

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

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

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

1.1 MySQL Enterprise Monitor Component Overview

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

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

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

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

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

**Figure 1.2 MySQL Enterprise Monitor Agentless Architecture**

![MySQL Enterprise Monitor Agentless Architecture](image)

---

**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.

**1.2 MySQL Enterprise Monitor Agent**

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

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

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

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

• Collects the data from the server, including obtaining the values and configuration of the MySQL server, status variables and other information.

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

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

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

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

1.3 MySQL Enterprise Service Manager

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

• Receiving and storing information from the Agents.

• Configuring the types of information collected by the Agents.

• Analyzing the collected data using the Advisors.

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

• Displaying the collected data, events and notifications.

• Graphing and reporting on the collected data.

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

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

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

MySQL Enterprise Service Manager Repository

The repository is a MySQL instance which stores all data collected by the Agent. The majority of the data collected by the Agent is analyzed on-the-fly by the Advisors, then stored in the repository. The graphs and reports 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:

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

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

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

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

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

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

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

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

MySQL Enterprise Advisors

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

The following types of Advisor are provided:

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

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

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

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

- **Cluster**: Checks the status of the monitored MySQL Cluster.
- **Graphing**: Data for graphs.
- **Memory Usage**: Indicate how efficiently you are using various memory caches, such as the InnoDB buffer pool, MyISAM key cache, query cache, table cache, and thread cache.
- **Monitoring and Support Services**: Advisors related to the MySQL Enterprise Monitoring services itself.
- **Operating System**: Checks the Host Operating System performance.
- **Performance**: Identifies potential performance bottlenecks, and suggests optimizations.
- **Query Analysis**: Advisors related to Queries and Query Analysis.
- **Replication**: Identifies replication bottlenecks, and suggests replication design improvements.
- **Schema**: Identifies schema changes.
- **Security**: Checks MySQL Servers for known security issues.

It is also possible to create custom Advisors.

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

For more information on advisors, see Chapter 23, *Expression-Based Advisor Reference* and Chapter 24, *GUI-Based Advisor Reference*.

Events and Notifications

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

- **Events**: If an Advisor's defined Threshold is breached, an Event is generated and displayed on the Events page. This is the default alert method.
- **Notifications**: MySQL Enterprise Service Manager can be configured to send alerts by e-mail, or SNMP traps. These methods must be configured and are not enabled by default.

Query Analyzer

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

- **Performance Schema**: for monitored versions of MySQL 5.6.14 or higher, the Agent retrieves query information from the Performance Schema.
- **MySQL Enterprise Monitor Aggregator**: aggregates raw query statistics taken directly from client connections, but analyzed out-of-band and transmitted to the MySQL Enterprise Service Manager saving memory and processing overhead for client statements. MySQL Enterprise Monitor Aggregator can provide data from the Connector/PHP or, when used with the MySQL Enterprise Monitor Proxy, directly from the client application.

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

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

For more information, see Chapter 12, Configuring Connectors

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

1.4 MySQL Enterprise Monitor Proxy and Aggregator

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

The MySQL Enterprise Monitor Aggregator requires a framework, or chassis, to handle the communications between the client application and MySQL instance, and to enable the MySQL Enterprise Monitor Aggregator to communicate with the MySQL Enterprise Service Manager. The following frameworks are available:

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

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

Important
Currently, the Aggregator is only required by the MySQL Enterprise Plugin for Connector/PHP. The other connectors can be configured to communicate query data with MySQL Enterprise Service Manager and do not require MySQL Enterprise Monitor Aggregator.
Chapter 2 What's New in 3.4

Table of Contents

2.1 Performance ................................................................. 11
2.2 Group Replication .......................................................... 11
2.3 Table Statistics Report ...................................................... 11

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

2.1 Performance

Timeseries graph data is now aggregated continuously. This enables faster performance for timeseries graphs. It is particularly noticeable for long-period graphs.

If you are updating from an earlier version, your existing timeseries data is automatically aggregated when you first start the Service Manager.

Additionally, the page-loading times and page responsiveness in the user interface are improved.

2.2 Group Replication

The Replication dashboard is extended to provide monitoring support for Group Replication, introduced in MySQL Server 5.7.17.

The Topology view provides a visual representation of your group replication topologies and the Status drilldowns are updated with Group Replication-specific information. The status of the entire group is reported on, including node-failure tolerances and whether Group Replication has quorum.

In addition, new 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 misconfigurations which could lead to unstable or insecure installations While the status advisor continuously monitors for servers which go offline, or fall out of sync with the other members of the group.

2.3 Table Statistics Report

You can now view detailed table statistics in both table and treemap forms, with the new Table Statistics report.

This report utilizes the schema_table_statistics view of the sys schema, which can be installed on the selected instance from within MySQL Enterprise Monitor on MySQL 5.6 and above, to provide details on total latencies, total number of rows deleted, total I/O write requests, and so on.

For more information, see Section 20.4, “Table Statistics”.
Part II Installing MySQL Enterprise Monitor
Table of Contents

3 Installation Prerequisites ......................................................... 17
  3.1 Installer Files ..................................................................... 17
  3.2 Prerequisites ..................................................................... 18
    3.2.1 System Requirements ...................................................... 18
    3.2.2 Supported Platforms ..................................................... 19
    3.2.3 MySQL Enterprise Monitor Repository ............................ 21
  3.3 Credentials Required for Installation .................................. 23
    3.3.1 Existing Users .............................................................. 23
    3.3.2 Users Created During Installation ................................. 23
    3.3.3 Users Created on First Log-in ....................................... 24
  3.4 Supported Browsers ......................................................... 24

4 Service Manager Installation ................................................... 25
  4.1 MySQL Enterprise Monitor Installation Types .................... 25
  4.2 MySQL Enterprise Service Manager Graphical Installation Wizard 26
  4.3 Text-Based Installation ....................................................... 29
  4.4 Starting/Stopping the MySQL Enterprise Monitor Services ....... 29
  4.5 MySQL Enterprise Service Manager Configuration Settings .... 31

5 Monitor Agent Installation ....................................................... 33
  5.1 General Agent Related Notes ............................................. 33
  5.2 Creating MySQL User Accounts for the Monitor Agent ........ 34
  5.3 Java Considerations on Linux ............................................. 37
  5.4 Monitoring Agent Graphical Installation Wizard ................. 38
  5.5 Starting/Stopping MySQL Enterprise Monitor Agent .......... 41
    5.5.1 Starting/Stopping the Agent on Windows ...................... 41
    5.5.2 Starting/Stopping the Agent on Mac OS X ..................... 42
    5.5.3 Starting/Stopping the Agent on Unix ........................... 43
    5.5.4 sql_mode ................................................................. 44
  5.6 Monitoring Multiple MySQL Servers ................................. 44
  5.7 Configuring an Agent to Monitor a Remote MySQL Server ..... 45
  5.8 Monitoring Outside the Firewall with an SSH Tunnel .......... 45
  5.9 HTTP Connection Timeout ................................................. 46
  5.10 Troubleshooting the Agent ................................................ 46
  5.11 Agent Backlog ............................................................... 48

6 Upgrading MySQL Enterprise Monitor Installations .................. 49
  6.1 General considerations when upgrading MySQL Enterprise Monitor 49
  6.2 Upgrading to MySQL Enterprise Monitor 3.4 ....................... 50
  6.3 Restoring from Backup ..................................................... 51

7 Post-installation Considerations .............................................. 53
  7.1 General Considerations ..................................................... 53
  7.2 Installing SSL Certificates ............................................... 54
  7.3 Backup the Repository ...................................................... 58
  7.4 Changing an SSH Host Key ............................................... 59

8 Unattended Installation Reference .......................................... 61
  8.1 Unattended Installation .................................................... 61
    8.1.1 Performing an Unattended Installation ......................... 61
    8.1.2 MySQL Enterprise Service Manager Options .................. 62
    8.1.3 MySQL Enterprise Monitor Agent Options .................... 69

9 Performance Tuning MySQL Enterprise Monitor ...................... 79
  9.1 Tuning Memory .............................................................. 79
  9.2 Tuning CPU .................................................................. 81
  9.3 Tuning Apache Tomcat Threads ......................................... 81
Chapter 3 Installation Prerequisites

Table of Contents

3.1 Installer Files ................................. 17
3.2 Prerequisites .................................. 18
   3.2.1 System Requirements ..................... 18
   3.2.2 Supported Platforms ...................... 19
   3.2.3 MySQL Enterprise Monitor Repository .... 21
3.3 Credentials Required for Installation ............ 23
   3.3.1 Existing Users .......................... 23
   3.3.2 Users Created During Installation .......... 23
   3.3.3 Users Created on First Log-in .............. 24
3.4 Supported Browsers ................................ 24

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

A working installation requires the following:

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

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

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

Note

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

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

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

3.1 Installer Files

The MySQL Enterprise Monitor files include:

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

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

### 3.2 Prerequisites

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

#### 3.2.1 System Requirements

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

**Minimum Hardware Requirements**

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

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

**Recommended Hardware Requirements**

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

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

**MySQL Enterprise Monitor Disk space Requirements**

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

<table>
<thead>
<tr>
<th>Platform</th>
<th>Service Manager Minimum Disk Space</th>
<th>Agent Minimum Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux x86 32-bit</td>
<td>N/A</td>
<td>600 MB</td>
</tr>
<tr>
<td>Linux x86 64-bit</td>
<td>1.3 GB</td>
<td>800 MB</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>1.2 GB</td>
<td>700 MB</td>
</tr>
<tr>
<td>Solaris x86 64-bit</td>
<td>1.8 GB</td>
<td>800 MB</td>
</tr>
<tr>
<td>Solaris Sparc 64-bit</td>
<td>1.7 GB</td>
<td>600 MB</td>
</tr>
</tbody>
</table>
### Supported Platforms

<table>
<thead>
<tr>
<th>Platform</th>
<th>Service Manager Minimum Disk Space</th>
<th>Agent Minimum Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeBSD</td>
<td>N/A</td>
<td>300 MB (the FreeBSD installation does not include a JRE. It is assumed a compatible JRE is present on the system.)</td>
</tr>
<tr>
<td>Windows x86 32-bit</td>
<td>N/A</td>
<td>500 MB</td>
</tr>
<tr>
<td>Windows x86 64-bit</td>
<td>800 MB</td>
<td>500 MB</td>
</tr>
</tbody>
</table>

**Important**

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

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

**Important**

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

### 3.2.2 Supported Platforms

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

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

For platform support updates, see MySQL Product Support Announcements.

### General Platform Recommendations

The following are recommended:

- Ensure that your Service Manager and Agent hosts are synchronized to the same time server. It is important that all times are properly synchronized.

- Ensure that your Service Manager and Agent hosts use different SSH host keys before installing.

- On Linux platforms, ensure you have both the `libaio` and `libnuma` libraries installed. The installer checks for both.
• The 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.

**Note**

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

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

Under Debian/Ubuntu:

```bash
shell> dpkg -l | grep -i lsb
ii  lsb-base                                   3.2-20ubuntu4
ii  lsb-release                                3.2-20ubuntu4
          Linux Standard Base 3.2 init script function
ii  lsb-release                                3.2-20ubuntu4
          Linux Standard Base version reporting utility
```

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

**MySQL Requirements**

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

• The MySQL Enterprise Service Manager installation includes the latest version of MySQL Server. If you intend to use a MySQL repository other than the one bundled in the MySQL Enterprise Service Manager installation, it is recommended that you use the latest MySQL 5.7.x version. The minimum supported version is MySQL Server 5.7.9.

**Important**

It is not possible to monitor MySQL 8 with this version of MySQL Enterprise Monitor, nor is it possible to configure this version of MySQL Enterprise Monitor to use MySQL 8 as its repository. To monitor MySQL 8, you must use MySQL Enterprise Monitor 8.0.

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

• The Monitoring Agent can monitor any version of MySQL Server from version 5.5 onwards.

**Important**

It is not possible to monitor pre-GA versions of MySQL 5.7. That is, MySQL versions 5.7.0 to 5.7.5 are not supported. MySQL Enterprise Monitor supports monitoring of MySQL 5.7.6 onwards.
• The monitoring Agent always uses `PERFORMANCE_SCHEMA.GLOBAL_STATUS` on MySQL 5.7 versions, and supports both modes of `show_compatibility_56` from MySQL 5.7.9 onwards.

**Note**

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

### 3.2.3 MySQL Enterprise Monitor Repository

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

**Important**

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

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

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

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

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

**Important**

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

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

Ensure the following:

• The MySQL Server version is 5.7.9 or higher.

**Important**

It is not possible to monitor MySQL 8 with this version of MySQL Enterprise Monitor, nor is it possible to configure this version of MySQL Enterprise Monitor to use MySQL 8 as its repository. To monitor MySQL 8, you must use MySQL Enterprise Monitor 8.0.

**Note**

It is not possible to use any MySQL version preceding 5.7.9, for the 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.
• SSL is enabled.

You must ensure the following in the MySQL Server configuration:
• Query Cache must not be enabled.
• Set `innodb_file_per_table=1`.
• Set `innodb_file_format=Barracuda`.
• On Linux/Unix hosts, ensure `innodb_flush_method=O_DIRECT`, except on Solaris if ZFS is used. If using ZFS, comment out this parameter.
• It is recommended to set `innodb_log_file_size=2048M`.
• Define a Service Manager user to enable the MySQL Enterprise Service Manager to connect to, and modify, the repository. This user must have the following privileges:
  • 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 the MySQL Enterprise Service Manager installation, the installer checks your MySQL Server for the following, minimum requirements:

• MySQL Server version: versions older then MySQL 5.7.9 are not supported.
• SSL: SSL must be enabled on the MySQL Server.
• `innodb_file_format`: `innodb_file_format` must be `barracuda`.
• `innodb_file_per_table`: `innodb_file_per_table` must be enabled.
• Solaris platforms: the Oracle Developer Studio 12.5 Runtime libraries must be installed.
  
  For more information, see [Installing MySQL on Solaris Using a Solaris PKG](#).

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

The upgrade process performs the same checks.
3.3 Credentials Required for Installation

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

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

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

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

3.3.1 Existing Users

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

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

3.3.2 Users Created During Installation

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

At the end of MySQL Enterprise Service Manager installation, the file configuration_report.txt is created, and along with other configuration details, contains some of the credentials of the repository manager. Look for this file in the following directories:

- **Windows**: C:\Program Files\MySQL\Enterprise\Monitor
- **Unix**: /opt/mysql/enterprise/monitor
- **Mac OS X**: /Applications/mysql/enterprise/monitor
### 3.3.3 Users Created on First Log-in

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

The **Agent user**: The Monitor Agent needs to report the status of the MySQL server it is monitoring. For this reason it needs to log in to the Monitor UI. You choose the ID and password for this user.

**Note**

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

### 3.4 Supported Browsers

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

- Microsoft Internet Explorer 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 4 Service Manager Installation

Table of Contents

4.1 MySQL Enterprise Monitor Installation Types ............................................................... 25
4.2 MySQL Enterprise Service Manager Graphical Installation Wizard ................................. 26
4.3 Text-Based Installation ................................................................................................. 29
4.4 Starting/Stopping the MySQL Enterprise Monitor Services ........................................... 29
4.5 MySQL Enterprise Service Manager Configuration Settings ......................................... 31

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

The MySQL Enterprise Service Manager installer installs the following components:

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

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

• MySQL Server: optional component. Used to store the data from the monitored hosts and instances. Referred to, throughout this document, as the repository. It is also possible to use another MySQL instance as the repository. For more information, see Section 3.2.3, “MySQL Enterprise Monitor Repository”.

4.1 MySQL Enterprise Monitor Installation Types

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

The following are the possible installation types:

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

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

• Large: More than 100 MySQL Servers monitored from a high-end server, dedicated to MySQL Enterprise Service Manager, with more than 8GB RAM.

These parameters are set in the following configuration files:

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

• my.cnf/my.ini:
  • table_definition_cache: defines the number of table definitions that can be stored in the definition cache.
MySQL Enterprise Service Manager Graphical Installation Wizard

- **innodb_buffer_pool_size**: defines the size, in megabytes, of the InnoDB buffer pool.

### Table 4.1 Installation Parameters

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

**Important**

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

### 4.2 MySQL Enterprise Service Manager Graphical Installation Wizard

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

**Note**

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

**Important**

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

To install MySQL Enterprise Service Manager, do the following:

1. Run the installer as required by the operating system.
2. The language selection dialog is displayed. Choose a language and click **OK**.
   
   The following information is displayed:
   
   **Note**
   
   During the installation process you must enter usernames and passwords for components of the Enterprise Monitor. Make note of these in a secure location so you can recover them in case they are forgotten.

3. Click **OK** to continue.
4. On the **Welcome** dialog, click **Forward**.
   
   The **Installation Directory** dialog is displayed.

5. Change the installation directory or accept the default path and click **Forward**.
The **Select Requirements** dialog is displayed.

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

   Click **Forward**.

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

7. Complete the following fields as required:

   • **Tomcat Server Port**: Default value is 18080.

   • **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 GRANT ALL PRIVILEGES ON *.* TO 'SM_UserName'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;

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

     **Important**

     It is not possible to configure this version of MySQL Enterprise Monitor to use MySQL 8 as its repository. To use MySQL 8 as repository, you must use MySQL Enterprise Monitor 8.0.

     **Note**

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

   Click **Forward**.

   The **Repository Configuration** dialog is displayed.
10. Complete the following fields:

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

  The default username is `service_manager`.

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

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

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

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

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

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

  Click **Forward**.

![Important](image)

**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 3.2.3, "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.

4.3 Text-Based Installation

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

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

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

1. Run the installer with the following option:

   ```
   --mode text
   ```

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

   ```
   shell>./mysqlmonitor-3.4.5.4244-linux-x86-64bit-installer.bin --mode text
   ```

   The text installation process starts.

2. Follow the instructions onscreen. The options and values are identical to those described in Section 4.2, “MySQL Enterprise Service Manager Graphical Installation Wizard”.

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

4.4 Starting/Stopping the MySQL Enterprise Monitor Services

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

The following services are installed by MySQL Enterprise Service Manager:

- MySQL Server
- Tomcat Server

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

The `help` parameter produces the following output:

```
shell>./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)
```
Starting/Stopping the MySQL Enterprise Monitor Services on Windows

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

To start the service:

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

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

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

**Important**

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

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

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

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

Starting/Stopping the MySQL Enterprise Monitor Services on Windows

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

- The `Start/Stop` MySQL Enterprise Monitor Services items on the Windows Start menu.

- The `Services` pane of the Microsoft Management Console. Right-click on the either of the MySQL Enterprise services to display the available options.

- The Windows command line, using the `sc` or `net` commands.

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

  For example:

  ```shell
  shell> sc start mysqlenterprisemysql
  ```

**Note**

The command line terminal must be started using the **Run as Administrator** option.
4.5 MySQL Enterprise Service Manager Configuration Settings

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

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

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

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

⚠️ Important

On first start, MySQL Enterprise Service Manager can take some time to start while the services and database initialize.
Chapter 5 Monitor Agent Installation

Table of Contents

5.1 General Agent Related Notes ................................................................. 33
5.2 Creating MySQL User Accounts for the Monitor Agent .......................... 34
5.3 Java Considerations on Linux ................................................................. 37
5.4 Monitoring Agent Graphical Installation Wizard ...................................... 38
5.5 Starting/Stopping MySQL Enterprise Monitor Agent .............................. 41
  5.5.1 Starting/Stopping the Agent on Windows ........................................... 41
  5.5.2 Starting/Stopping the Agent on Mac OS X ........................................ 42
  5.5.3 Starting/Stopping the Agent on Unix ............................................... 43
  5.5.4 sql_mode .................................................................................. 44
5.6 Monitoring Multiple MySQL Servers .................................................... 44
5.7 Configuring an Agent to Monitor a Remote MySQL Server ...................... 45
5.8 Monitoring Outside the Firewall with an SSH Tunnel .............................. 45
5.9 HTTP Connection Timeout ...................................................................... 46
5.10 Troubleshooting the Agent ...................................................................... 46
5.11 Agent Backlog ....................................................................................... 48

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

⚠ Important

It is not possible to configure this version of MySQL Enterprise Monitor Agent to monitor MySQL 8. To monitor MySQL 8, you must use MySQL Enterprise Monitor 8.0.

⚠ Important

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

It is strongly recommended that MySQL Enterprise Monitor Agent 3.4 are installed for use with the MySQL Enterprise Service Manager 3.4.

5.1 General Agent Related Notes

This section describes important features of the Agent.

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

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

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

• Choosing a MySQL Instance to monitor during the installation is optional. If you choose to define a MySQL Instance while running the Installer, additional MySQL Instances on the host are detected and
Creating MySQL User Accounts for the Monitor Agent

reported in the 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.

• Proxy and Aggregator

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

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

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

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

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

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

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

5.2 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, it should have 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 (which with large numbers of databases and tables can become costly), or any custom SQL that is used to monitor application specific statistics.

If you are manually managing this user, it should have at least the `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.
Creating the Admin User

- **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.

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

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

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

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

```sql
GRANT CREATE, INSERT
ON mysql.*
```
Creating the Limited and General Users

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:

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

Note
Perform this step after running the agent on the corresponding MySQL server to ensure that the `mysql.inventory` table is created correctly. 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 `SET SQL_LOG_BIN=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 5.7, “Configuring an Agent to Monitor a Remote MySQL Server”.

5.3 Java Considerations on Linux

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

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

Compatibility Libraries

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

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

5.4 Monitoring Agent Graphical Installation Wizard

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

Important
It is not possible to configure this version of MySQL Enterprise Monitor Agent to monitor MySQL 8. To monitor MySQL 8, you must use MySQL Enterprise Monitor 8.0.

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 8.1, “Unattended Installation”.

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

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

### 5.5 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.

#### 5.5.1 Starting/Stopping the Agent on Windows

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

#### Note

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

#### Warning

To report its findings, the agent needs to be able to connect to the Monitor UI through the port specified during installation. The default value for this port is **18443**; ensure that this port is not blocked. If you need help troubleshooting the agent installation see, **Section 5.10, “Troubleshooting the Agent”**.

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

```
shell> sc start MySQLEnterpriseMonitorAgent
```

or:

```
Starting/Stopping the Agent on Mac OS X

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

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

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

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

```
shell> sc stop MySQLEnterpriseMonitorAgent
```

or:

```
shell> net stop MySQLEnterpriseMonitorAgent
```

Note

MySQLEnterpriseMonitorAgent is the default name of the Monitor Agent service.

5.5.2 Starting/Stopping the Agent on Mac OS X

Using launchd

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

To start (load) the Agent:

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

To stop (unload) the Agent:

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

Using init

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

```
shell> ./mysql-monitor-agent start
```
To stop the Agent, use the `stop` command:

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

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

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

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

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

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

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

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

For further information on troubleshooting the agent, see Section 5.10, “Troubleshooting the Agent”.

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

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

```
shell> ./mysql-monitor-agent {start|stop|restart|status}
```

**Warning**

To report its findings, the agent connects to the Monitor UI through the port specified during installation. The default value for this port is **18443**; ensure that this port is not blocked. To troubleshoot the agent installation, see Section 5.10, “Troubleshooting the Agent”.

### 5.5.3 Starting/Stopping the Agent on Unix

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

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

For a non-root installation the command would be:

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

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

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

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

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

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

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

For further information on troubleshooting the agent, see Section 5.10, "Troubleshooting the Agent".

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

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

```
shell> ./mysql-monitor-agent {start|stop|restart|status}
```

Warning

To report its findings, the agent connects to the Monitor UI through the port specified during installation. The default value for this port is 18443; ensure that this port is not blocked. To troubleshoot the agent installation, see Section 5.10, "Troubleshooting the Agent".

### 5.5.4 sql_mode

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

### 5.6 Monitoring Multiple MySQL Servers

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

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

Typically, an Agent scans a host and reports unmonitored MySQL instances to MySQL Enterprise Monitor User Interface. For more information, see Section 1.2, "MySQL Enterprise Monitor Agent". For information
about how to change the status of a MySQL instance from unmonitored to monitored, see Chapter 16, *MySQL Instances Dashboard*.

### 5.7 Configuring an Agent to Monitor a Remote MySQL Server

Typically, the Agent runs on the same machine as the MySQL servers that it is monitoring. To monitor MySQL servers running on remote hosts, you can install the Agent on a machine other than the one hosting the MySQL server.

The process for installing an Agent to monitor a MySQL server on a remote machine is identical to the process described in Chapter 5, *Monitor Agent Installation*. Follow the directions given there, being careful to either select "host-only" and add remote MySQL instances later, or specify the correct IP address or host name for MySQL Enterprise Service Manager and likewise for the MySQL server—since the Agent is not running on the same machine as the MySQL server, it cannot be the default (*localhost*).

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

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

#### Remote Monitoring Limitations

- Remote monitoring does not provide operating system level data, such as CPU, file, and network utilization information.

- Monitoring multiple MySQL instances with a single agent potentially means having a single point of failure. This is especially true for remote monitoring, as it might lose a connection, which means a black period, whereas a local Agent continues monitoring and provides information upon reconnection.

- For replication autodiscovery, do not use remote monitoring with replication slaves or masters. The Agent must be installed on the same machine as the server you are monitoring in order for discovery to work properly. For more information, see Chapter 19, *Replication Dashboard*.

### 5.8 Monitoring Outside the Firewall with an SSH Tunnel

If you run an SSH server on the machine that hosts the MySQL Enterprise Service Manager and an SSH client on the machine that hosts the agent, you can create an SSH tunnel so that the agent can bypass your firewall. First, you need to make an adjustment to the `agent-mgmt-hostname` value specified in the *etc/bootstrap.properties* configuration file. Stop the agent and change the `hostname` value as shown in the following:

```plaintext
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:

```sh
shell> ssh -L 18443:Monitor_UI_Host:18443 -l user_name -N Monitor_UI_Host
```
When prompted, enter the password for `user_name`.

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

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

- Under Windows see, Section 5.5.1, “Starting/Stopping the Agent on Windows”.
- Under Unix see, Section 5.5.3, “Starting/Stopping the Agent on Unix”.
- Under Mac OS X see, Section 5.5.2, “Starting/Stopping the Agent on Mac OS X”.

### 5.9 HTTP Connection Timeout

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

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

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

### 5.10 Troubleshooting the Agent

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

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

Some additional tips are:

- To run on start-up, the agent requires correct login credentials for the monitored MySQL server. Log in to the monitored MySQL server and check the agent's credentials. Compare the values of the `Host`, and `User` fields in the `mysql.user` table with the values shown in the etc/agentManaged/mysqlConnection<id>/bean/json file. The passwords are encrypted so they can not be manually managed here, but the password can be altered from the MySQL Instances page in the MySQL Enterprise Monitor User Interface, or by using the agent connection tool (`agent.sh`) from the command line.

- Using incorrect credentials for logging in to the service manager creates an entry in the agent log file.

- An easy way to confirm that the agent can log in to the service manager is to type `https://Monitor_UI_Host:18443/heartbeat` into the address bar of your web browser, substituting the appropriate host name and port. When the HTTP authentication dialog box opens, enter the agent user name and password. The following message indicates a successful login:
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 file. 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 5.8, "Monitoring Outside the Firewall with an SSH Tunnel".

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

- If you have more than one agent running on the same machine, the UUID must be unique.

- If the agent and the MySQL server it is monitoring are running on different machines, ensure that the correct host is specified for the agent account. The correct port, typically 3306, must also be open for remote login. For more information about remote monitoring see, Section 5.7, "Configuring an Agent to Monitor a Remote MySQL Server".

- 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.
5.11 Agent Backlog

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

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

- Monitoring 10 MySQL instances: the agent backlog can store up to 4 minutes of monitored data before the backlog cache is filled and data dropped.
Chapter 6 Upgrading MySQL Enterprise Monitor Installations

Table of Contents

6.1 General considerations when upgrading MySQL Enterprise Monitor ................................. 49
6.2 Upgrading to MySQL Enterprise Monitor 3.4 ................................................................. 50
6.3 Restoring from Backup ....................................................................................................... 51

This chapter describes the upgrade process to MySQL Enterprise Monitor 3.4.

Important

It is not possible to upgrade any version older than MySQL Enterprise Monitor 3.1.0 directly to 3.4. To upgrade an older installation, you must first upgrade to 3.1.x, then upgrade to 3.4.

The upgrade installer generates an error if you attempt to upgrade an older version to 3.4.

6.1 General considerations when upgrading MySQL Enterprise Monitor

You cannot use the update installers to change to a different operating system or chip architecture. For example, you cannot update a 32-bit Linux installation to a 64-bit version using an update installer. You must perform a fresh installation instead.

Customizations to setenv.sh are lost, as this file is replaced and optimized 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 3.4 is bundled with MySQL 5.7.20. If you are using the bundled MySQL Server for your repository (default installation), the upgrade process upgrades your repository to 5.7.20. If you are using another MySQL Server as your repository, you must upgrade it manually before installing MySQL Enterprise Service Manager 3.4. For external repositories, you must use MySQL Server 5.7.9, or higher. 5.7.20 is recommended.
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.

6.2 Upgrading to MySQL Enterprise Monitor 3.4

To upgrade from 3.x to 3.4, you must upgrade your MySQL Enterprise Service Manager installation first, and your agents after the MySQL Enterprise Service Manager installation completes successfully.

Important

It is not possible to upgrade to MySQL Enterprise Monitor 3.4 from versions preceding MySQL Enterprise Monitor 3.1. To upgrade an older version, you must first upgrade to MySQL Enterprise Monitor 3.1 before running the MySQL Enterprise Monitor 3.4.0 upgrade process.

The upgrade installer generates an error if you attempt to upgrade an older version to 3.4.

Upgrade Installer

The name of the upgrade file varies, but includes the target operating system, the version installed by the upgrade, and the component name. For example, a file named `mysqlenterprisemanager-3.4.0-windows-update-installer.exe` updates MySQL Enterprise Service Manager on Windows to version 3.4.0.

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

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

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

Service Manager Upgrade Process

Important

If using an external repository, you must upgrade it to at least MySQL Server 5.7.9 before upgrading your MySQL Enterprise Service Manager. 5.7.18 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 to 3.4**

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.

**6.3 Restoring from Backup**

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

If you choose to back up your current installation, the installer checks that there is adequate disk space for your repository backup. If there is not enough space, you are given the option of choosing another location; you can also choose not to back up the repository.
Chapter 7 Post-installation Considerations

Table of Contents

7.1 General Considerations ........................................................................................................... 53
7.2 Installing SSL Certificates .................................................................................................... 54
7.3 Backup the Repository ......................................................................................................... 58
7.4 Changing an SSH Host Key ................................................................................................. 59

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

7.1 General Considerations

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

New Users

1. **Groups and Connections**: Groups have always been used to define Event handling and Advisor scheduling policies; in this release Groups can also be used to restrict visibility and access to specific MySQL instances and their hosts. Before you create Connections and set up Groups, we recommend you first read the note immediately following on Users, Roles, and Access Control

   • To create a Connection, select **MySQL Instances** from the **Dashboard** menu. Create new monitoring connections either by processing the unmonitored instances already discovered by MEM or by manually specifying connection parameters for each MySQL Instance you want to monitor. See Section 16.2, “MySQL Instance Details” for more information on creating connections in the User Interface.

   • Use the new Group Editor on the Settings menu to collect your MySQL instances into Groups.

2. **Users, Roles, and Access Control (ACLs)**: Before using the Settings menu to create User accounts, see Chapter 25, Access Control and Chapter 26, 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 17, Managing Groups of Instances. Only after you create groups can you create group-specific Roles.

   Finally, assign users to your Roles.

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

3. **Configure Event Handling and Notification policies**: Open **Event Handling** from the Settings menu. Complete, and test, the SMTP, or SNMP, configuration. See Chapter 22, Events and Event Handlers for more information.

4. **Overview Dashboard**: Select **Overview** from the **Dashboard** menu. Set the defaults for the groups you want to view, the time range, and graphs to display. See Chapter 15, Overview for more information.

5. **Replication Dashboard**: If you are using MySQL Replication, select the Dashboards menu, click on Replication and select a group to view its topology, configuration, status and replication error details. See Chapter 19, Replication Dashboard for more information.
6. **Advisors**: You can accept the defaults defined, or select Advisors from the Settings menu and customize the threshold for groups, or individual MySQL Instances. For more information, see Chapter 21, 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. (If you are monitoring earlier MySQL versions, make sure to download a Query Analyzer plug-in so you can see SQL performance data as well.)

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 new 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 20.2, “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 6, Upgrading MySQL Enterprise Monitor Installations.

- **Users, Roles, and Access Control (ACLs)**: This release introduces Access Control Lists (ACLs). Built-in Roles have replaced the privileges previously defined in Manage Users in version 3.0. Your system already has been migrated, however we strongly recommend reviewing Chapter 25, Access Control and Chapter 26, Access Control - Best Practices.

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

If intend to use ACLs to restrict visibility or access, we suggest you review your existing Groups with that in mind. The new Group Editor is located in the Settings menu. For an explanation of how privileges are migrated from earlier versions, see Section 25.6, “Default Users and Roles”.

- **Overview Dashboard**: If you haven’t already done so, select the Dashboards menu, click on Overview and set defaults for which Group you want to view, the graph time range, and the set and order of Graphs to display. See Chapter 17, Managing Groups of Instances.

- **SQL Performance Tuning**: If you are monitoring instances of MySQL running version 5.6.14 or later, make sure to see the rich SQL performance tuning data available in the Query Analyzer. (If you are monitoring earlier MySQL versions, make sure to download a Query Analyzer plug-in so you can see SQL performance data as well.)

- **I/O and Lock Contention**: If you are using MySQL 5.6 or later consider deploying the sys schema, and making use of the new Database File I/O and Lock Waits reports from the Reports & Graphs menu. These 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 20.2, “Database File I/O and Lock Waits” for more information.

---

**7.2 Installing SSL Certificates**

---

**Important**

The self-signed certificates are generated by the MySQL Enterprise Monitor installation or upgrade process, and are set to expire after 365 days. In the unlikely event you are running a version of MySQL Enterprise Service Manager using the
Checking the Keystore

default certificates for more than a year, you must generate new certificates. If you do not generate new certificates, the SSL connection between MySQL Enterprise Service Manager and the repository fails. This section describes how to generate those certificates.

These instructions guide you through the process of installing SSL certificates for your MySQL Enterprise Monitor installation.

Checking the Keystore

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

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

Generating Keystore, Key, and Certificate

For information on using keytool, see Java Keytool.

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

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

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

Important

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

MySQL Enterprise Service Manager SSL Import

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

To import your certificate, run the following command:

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

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

```
```

Important

The certificate name must be set to tomcat, or match the name used in the key generation steps.
To import the converted certificate, run the following command:

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

**Note**

If you are replacing an existing certificate with a new certificate using the same name, you must delete the existing certificate before importing the new one.

For example, if replacing a certificate named `tomcat`, which is the default in MySQL Enterprise Monitor, you must run the keytool with `-delete -name -tomcat` before running the import command.

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

**SSL for the Repository**

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

**MySQL Enterprise Monitor Agent**

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

**Table 7.1 Agent SSL Configuration Options**

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Password for the CA keystore, if <code>ssl-allow-self-signed-certs</code> is true.</td>
<td></td>
</tr>
</tbody>
</table>

An example `bootstrap.properties` SSL certification section:

```ini
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:

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

The tool responds with the certificate details. For example:

Enter keystore password: (the keystore 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
Certificate fingerprints:
Signature algorithm name: MD5withRSA
Version: 1
Trust this certificate? [no]: (type yes + enter)
Certificate was added to keystore

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

### LDAP SSL Configuration

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

**Important**

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

To connect to LDAP servers which implement the AES256 cipher, you must download and install the [Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8](https://en.wikipedia.org/wiki/Java_Cryptography_Extension) package. This package is available from: [Java Cryptography Extension](https://www.oracle.com/technetwork/java/javase/downloads/jce-8-download-2133166.html).
Backup the Repository

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

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

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

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

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

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

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

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

7.3 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`:

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

The above command creates a file, `mem.dump`, containing all of the MySQL Enterprise Monitor data.

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

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

You should also backup the following files:

- `apache-tomcat/conf/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`
Changing an SSH Host Key

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 the MySQL Enterprise Service Manager upgrade installer.

### 7.4 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.
- Edit the `hostid` in the MySQL Enterprise Service Manager repository.

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

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

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

   ```
   shell> ssh-keygen -t rsa -N '' -f /etc/ssh/ssh_host_key
   ```

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

   **Note**

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

2. Retrieve the key fingerprint.

   The fingerprint is an alphanumeric string similar to the following:


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

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

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

3. Stop the monitoring agent.
4. Open the monitoring agent's `bootstrap.properties` configuration file, and add, or edit, the following value:

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

For example, using the fingerprint listed above:

```
```

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

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

6. Restart the monitoring agent.
Chapter 8 Unattended Installation Reference

Table of Contents

8.1 Unattended Installation .......................................................... 61
   8.1.1 Performing an Unattended Installation ................................ 61
   8.1.2 MySQL Enterprise Service Manager Options ................................. 62
   8.1.3 MySQL Enterprise Monitor Agent Options .................................. 69

8.1 Unattended Installation

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

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

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

8.1.1 Performing an Unattended Installation

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

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

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

```
shell> mysqlmonitoragent-version-linux-x86-64bit-installer.bin
   --installdir /data0/mysql/agent
   --mysqlhost 127.0.0.1 --mysqlport 3306
   --mysqluser root --mysqlpassword foo --agent_autocreate
   --limiteduser limited --limitedpassword foo --generaluser general --generalpassword foo
   --checkmysqlhost yes --managerhost localhost --managerport 48080 --agentuser AGENTUSER
   --agentpassword PASSWORD --mode unattended --mysql-identity-source default
```

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

```
debugtrace=/opt/mysql/enterprise/install.debugtrace.monitor.log
mode=unattended
installdir=/opt/mysql/enterprise/monitor
tomcatport=8080
tomcatsslport=8443
```
This file identifies a directory and file name for a log file, sets the `mode` to `unattended`, and uses the `installdir` option to specify an installation directory.

**Note**

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

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

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

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

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

On Windows within a command shell:

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

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

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

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

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

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

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

### 8.1.2 MySQL Enterprise Service Manager Options

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

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--adminpassword</code></td>
<td>Password for the database repository</td>
<td></td>
</tr>
<tr>
<td><code>--adminuser</code></td>
<td>Username for the database repository</td>
<td></td>
</tr>
<tr>
<td><code>--backupdir</code></td>
<td>Backup directory path</td>
<td></td>
</tr>
</tbody>
</table>
## MySQL Enterprise Service Manager Options

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>--createDataBackup</td>
<td>Backup stored data. Upgrade process only.</td>
<td></td>
</tr>
<tr>
<td>--dbhost</td>
<td>Hostname or IP address of the MySQL server</td>
<td></td>
</tr>
<tr>
<td>--dbname</td>
<td>Name of the repository database.</td>
<td></td>
</tr>
<tr>
<td>--dbport</td>
<td>TCP/IP port for the MySQL server</td>
<td></td>
</tr>
<tr>
<td>--debuglevel</td>
<td>Set the debug information level</td>
<td></td>
</tr>
<tr>
<td>--debugtrace</td>
<td>File for a debug trace of the installation</td>
<td></td>
</tr>
<tr>
<td>--forceRestart</td>
<td>Upgrade only. Restarts the services after the upgrade process completes.</td>
<td></td>
</tr>
<tr>
<td>--help</td>
<td>Display the list of valid options</td>
<td></td>
</tr>
<tr>
<td>--installdir</td>
<td>Installation directory</td>
<td></td>
</tr>
<tr>
<td>--installer-language</td>
<td>Language selection</td>
<td></td>
</tr>
<tr>
<td>--mode</td>
<td>Installation mode</td>
<td></td>
</tr>
<tr>
<td>--mysql_installation_type</td>
<td>MySQL server to be used by the MySQL Enterprise Monitor</td>
<td></td>
</tr>
<tr>
<td>--mysql_ssl</td>
<td>Use SSL when connecting to the database</td>
<td>3.4.3</td>
</tr>
<tr>
<td>--optionfile</td>
<td>Installation option file</td>
<td></td>
</tr>
<tr>
<td>--system_size</td>
<td>Defines Tomcat and MySQL repository configuration based on installation size.</td>
<td></td>
</tr>
<tr>
<td>--tomcatport</td>
<td>Server port for the Tomcat component</td>
<td></td>
</tr>
<tr>
<td>--tomcatsslport</td>
<td>SSL TCP/IP port for the Tomcat component</td>
<td></td>
</tr>
<tr>
<td>--unattendedmodeui</td>
<td>Unattended mode user interface</td>
<td></td>
</tr>
<tr>
<td>--version</td>
<td>Display the product information</td>
<td></td>
</tr>
</tbody>
</table>

- **--help**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--help</td>
</tr>
</tbody>
</table>

Display the list of valid installer options.

- **--version**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--version</td>
</tr>
</tbody>
</table>

Display product and version information.

- **--backupdir**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--backupdir</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

Upgrade only. The backup directory.
MySQL Enterprise Service Manager Options

- **--createDataBackup**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--createDataBackup</td>
</tr>
<tr>
<td>Type</td>
<td>Boolean</td>
</tr>
<tr>
<td>Default Value</td>
<td>1</td>
</tr>
<tr>
<td>Valid Values</td>
<td>0 (Do not create data backup)</td>
</tr>
<tr>
<td></td>
<td>1 (Create data backup)</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--optionfile</td>
</tr>
</tbody>
</table>

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

- **--mode**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mode</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value (Windows)</td>
<td>win32</td>
</tr>
<tr>
<td>Default Value (Unix)</td>
<td>xwindow</td>
</tr>
<tr>
<td>Default Value (OS X)</td>
<td>osx</td>
</tr>
<tr>
<td>Default Value (Linux)</td>
<td>gtk</td>
</tr>
<tr>
<td>Valid Values (Windows)</td>
<td>win32 (Windows (native))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
<tr>
<td>Valid Values (Unix)</td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
<tr>
<td>Valid Values (OS X)</td>
<td>osx (Mac OS X (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
<tr>
<td>Valid Values (Linux)</td>
<td>gtk (GTK (X Windows))</td>
</tr>
<tr>
<td></td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>
MySQL Enterprise Service Manager Options

The installation mode to use for this installation.

- **--debugtrace**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--debugtrace</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

- **--debuglevel**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--debuglevel</td>
</tr>
<tr>
<td>Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default Value</td>
<td>2</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>4</td>
</tr>
</tbody>
</table>

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

- **--installer-language**

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

The installer language.

- **--installdir**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--installdir</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value (Windows)</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor</td>
</tr>
<tr>
<td>Default Value (Unix)</td>
<td>/opt/mysql/enterprise/monitor/</td>
</tr>
<tr>
<td>Default Value (OS X)</td>
<td>/Applications/mysql/enterprise/monitor/</td>
</tr>
</tbody>
</table>

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.
MySQL Enterprise Service Manager Options

- **--system-size**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--system-size</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>medium</td>
</tr>
<tr>
<td>Valid Values</td>
<td>small (5 to 10 MySQL Servers monitored from a laptop or low-end server with no more than 4GB of RAM.)</td>
</tr>
<tr>
<td></td>
<td>medium (Up to 100 MySQL Servers monitored from a medium-sized, but shared, server with 4 to 8GB of RAM.)</td>
</tr>
<tr>
<td></td>
<td>large (More than 100 MySQL Servers monitored from a high-end, dedicated server, with more than 8GB RAM.)</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--tomcatport</td>
</tr>
<tr>
<td>Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default Value</td>
<td>18080</td>
</tr>
</tbody>
</table>

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

- **--tomcatsslport**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--tomcatsslport</td>
</tr>
<tr>
<td>Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default Value</td>
<td>18443</td>
</tr>
</tbody>
</table>

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

- **--mysql-identity-source**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysql-identity-source</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>default</td>
</tr>
<tr>
<td>Valid Values</td>
<td>default (Default)</td>
</tr>
</tbody>
</table>
### MySQL Enterprise Service Manager Options

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_plus_datadir</td>
<td>(host_plus_datadir)</td>
</tr>
</tbody>
</table>

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

**Note**

This option is only available in unattended installation mode.

**Note**

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

- **--adminuser**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td><code>--adminuser</code></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td><code>service_manager</code></td>
</tr>
</tbody>
</table>

The user name to use for connecting to the database repository used by MySQL Enterprise Service Manager. If you install the bundled MySQL server, this user is configured in the new database. If you use an existing MySQL server, specify an existing user with rights to access the database.

**Note**

The repository user name and encrypted password are stored in the `config.properties` configuration file.

- **--unattendedmodeui**

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

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

- **--adminpassword**
### MySQL Enterprise Service Manager Options

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td><code>--adminpassword</code></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

- **`--mysql_installation_type`**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td><code>--mysql-installation-type</code></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td><code>bundled</code></td>
</tr>
<tr>
<td>Valid Values</td>
<td>&quot;bundled&quot; (Use the bundled MySQL server)</td>
</tr>
<tr>
<td></td>
<td>&quot;existing&quot; (Use an existing (user supplied) MySQL server)</td>
</tr>
</tbody>
</table>

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

- **`--dbport`**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td><code>--dbport</code></td>
</tr>
<tr>
<td>Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default Value</td>
<td>13306</td>
</tr>
</tbody>
</table>

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

- **`--dbhost`**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td><code>--dbhost</code></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>127.0.0.1</td>
</tr>
</tbody>
</table>

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`**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td><code>--dbname</code></td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>
MySQL Enterprise Monitor Agent Options

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>mem</td>
</tr>
</tbody>
</table>

The name of the MySQL Enterprise Service Manager repository.

- `--forceRestart`

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td><code>--forceRestart</code></td>
</tr>
<tr>
<td>Type</td>
<td>Boolean</td>
</tr>
<tr>
<td>Default Value</td>
<td>0</td>
</tr>
<tr>
<td>Valid Values</td>
<td>0 (Do not restart services) 1 (Restart services)</td>
</tr>
</tbody>
</table>

Force a restart of MySQL Enterprise Service Manager services.

### 8.1.3 MySQL Enterprise Monitor Agent Options

To view all the options available for an unattended agent installation, invoke the agent installer file passing in the `help` option. The available options are detailed in the following table.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--agent_installtype</code></td>
<td>Installation type for the agent, which can be database or standalone.</td>
</tr>
<tr>
<td><code>--agent_autocreate</code></td>
<td>Create an account on the monitored MySQL server to be used by the agent</td>
</tr>
<tr>
<td><code>--agentpassword</code></td>
<td>Password of the agent user for connecting to the monitored MySQL server</td>
</tr>
<tr>
<td><code>--agentservicename</code></td>
<td>Service name for the Agent</td>
</tr>
<tr>
<td><code>--agentuser</code></td>
<td>Username of the agent for connecting to the monitored MySQL server</td>
</tr>
<tr>
<td><code>--checkmysqlhost</code></td>
<td>Validate the supplied MySQL hostname</td>
</tr>
<tr>
<td><code>--createBackup</code></td>
<td>(Upgrade only) Create backup.</td>
</tr>
<tr>
<td><code>--debuglevel</code></td>
<td>Set the debug information level</td>
</tr>
<tr>
<td><code>--debugtrace</code></td>
<td>File for a debug trace of the installation</td>
</tr>
<tr>
<td><code>--generalpassword</code></td>
<td>General user password for the <code>--generaluser</code></td>
</tr>
<tr>
<td><code>--generaluser</code></td>
<td>General user username for the monitored MySQL server</td>
</tr>
<tr>
<td><code>--help</code></td>
<td>Display the list of valid options</td>
</tr>
<tr>
<td><code>--installdir</code></td>
<td>Installation directory</td>
</tr>
<tr>
<td><code>--installer-language</code></td>
<td>Language selection</td>
</tr>
<tr>
<td><code>--limitedpassword</code></td>
<td>Limited user password for the <code>--limiteduser</code></td>
</tr>
<tr>
<td><code>--limiteduser</code></td>
<td>Limited user username for the monitored MySQL server</td>
</tr>
<tr>
<td><code>--managerhost</code></td>
<td>Hostname of IP address of the MySQL Enterprise Monitor server</td>
</tr>
</tbody>
</table>
**MySQL Enterprise Monitor Agent Options**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--managerport</td>
<td>TCP/IP port of the MySQL Enterprise Monitor server</td>
</tr>
<tr>
<td>--mode</td>
<td>Installation mode</td>
</tr>
<tr>
<td>--mysql-identity-source</td>
<td>MySQL instance identify definition</td>
</tr>
<tr>
<td>--mysqlconnectiongroup</td>
<td>Sets the group for the provided MySQL connection</td>
</tr>
<tr>
<td>--mysqlconnexionmethod</td>
<td>Connection method to the monitored MySQL server</td>
</tr>
<tr>
<td>--mysqlhost</td>
<td>MySQL hostname/IP address</td>
</tr>
<tr>
<td>--mysqlpassword</td>
<td>MySQL password for the monitored --mysqluser.</td>
</tr>
<tr>
<td>--mysqlport</td>
<td>TCP/IP port for the monitored MySQL server</td>
</tr>
<tr>
<td>--mysqlsocket</td>
<td>Unix socket/Named pipe for the monitored MySQL server</td>
</tr>
<tr>
<td>--mysqluser</td>
<td>MySQL Administrative username for the monitored MySQL server</td>
</tr>
<tr>
<td>--optionfile</td>
<td>Installation option file</td>
</tr>
<tr>
<td>--restartImmediately</td>
<td>(Upgrade only) Restart Agent immediately after updating all files.</td>
</tr>
<tr>
<td>--unattendedmodeui</td>
<td>Unattended mode user interface</td>
</tr>
<tr>
<td>--version</td>
<td>Display the product information</td>
</tr>
</tbody>
</table>

- **--agentpassword**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--agentpassword</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

- **--createBackup**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--createBackup</td>
</tr>
<tr>
<td>Type</td>
<td>Boolean</td>
</tr>
<tr>
<td>Default Value</td>
<td>1</td>
</tr>
</tbody>
</table>

Whether to backup the data.

**Note**

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

- **--restartImmediately**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--restartImmediately</td>
</tr>
<tr>
<td>Type</td>
<td>Boolean</td>
</tr>
<tr>
<td>Default Value</td>
<td>1</td>
</tr>
</tbody>
</table>

Restart Agent immediately after updating all files.
MySQL Enterprise Monitor Agent Options

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

- **--agentuser**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--agentuser</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>agent</td>
</tr>
</tbody>
</table>

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

- **--checkmysqlhost**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--checkmysqlhost</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>yes</td>
</tr>
<tr>
<td>Valid Values</td>
<td>yes (Check host)</td>
</tr>
<tr>
<td></td>
<td>no (Do not check host)</td>
</tr>
</tbody>
</table>

Validate the MySQL hostname or IP address.

- **--debuglevel**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--debuglevel</td>
</tr>
<tr>
<td>Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default Value</td>
<td>2</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>4</td>
</tr>
</tbody>
</table>

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

- **--debugtrace**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--debugtrace</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

- **--installdir**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--installdir</td>
</tr>
</tbody>
</table>
MySQL Enterprise Monitor Agent Options

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value (Windows)</td>
<td>C:\Program Files\MySQL\Enterprise\Agent</td>
</tr>
<tr>
<td>Default Value (Unix)</td>
<td>/opt/mysql/enterprise/agent/</td>
</tr>
<tr>
<td>Default Value (OS X)</td>
<td>/Applications/mysql/enterprise/agent/</td>
</tr>
</tbody>
</table>

Specify the directory into which to install the software.

- **--installer-language**

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

Set the language to use for the installation process.

- **--managerhost**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--managerhost</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

- **--managerport**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--managerport</td>
</tr>
<tr>
<td>Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default Value</td>
<td>18443</td>
</tr>
</tbody>
</table>

Tomcat SSL Port

- **--mode**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mode</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value (Windows)</td>
<td>win32</td>
</tr>
<tr>
<td>Default Value (Unix)</td>
<td>xwindow</td>
</tr>
<tr>
<td>Default Value (OS X)</td>
<td>osx</td>
</tr>
<tr>
<td>Default Value (Linux)</td>
<td>gtk</td>
</tr>
</tbody>
</table>
**MySQL Enterprise Monitor Agent Options**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values (Windows)</td>
<td>win32 (Windows (native))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
<tr>
<td>Valid Values (Unix)</td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
<tr>
<td>Valid Values (OS X)</td>
<td>osx (Mac OS X (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
<tr>
<td>Valid Values (Linux)</td>
<td>gtk (GTK (X Windows))</td>
</tr>
<tr>
<td></td>
<td>xwindow (X Windows (native))</td>
</tr>
<tr>
<td></td>
<td>text (Text (command-line))</td>
</tr>
<tr>
<td></td>
<td>unattended (Unattended (no dialogs/prompts))</td>
</tr>
</tbody>
</table>

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

- **--mysqlconnmethod**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysqlconnmethod</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>tcpip</td>
</tr>
<tr>
<td>Valid Values</td>
<td>tcpip (Use TCP/IP)</td>
</tr>
<tr>
<td></td>
<td>socket (Use Unix Socket/Named Pipe)</td>
</tr>
</tbody>
</table>

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

**Note**

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

- **--mysqlhost**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysqlhost</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>127.0.0.1</td>
</tr>
</tbody>
</table>

Hostname or IP address of the MySQL server to be monitored.
MySQL Enterprise Monitor Agent Options

• **--mysqlpassword**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysqlpassword</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

• **--mysqlport**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysqlport</td>
</tr>
<tr>
<td>Type</td>
<td>Numeric</td>
</tr>
<tr>
<td>Default Value</td>
<td>3306</td>
</tr>
</tbody>
</table>

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

• **--mysqlsocket**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysqlsocket</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

• **--mysqluser**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysqluser</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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

• **--agent_autocreate**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--agent-autocreate</td>
</tr>
<tr>
<td>Type</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--generaluser</td>
</tr>
</tbody>
</table>

|
### MySQL Enterprise Monitor Agent Options

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

The username for the general user.

- **--generalpassword**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--generalpassword</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

Password for the **--generaluser**.

- **--limiteduser**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--limiteduser</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

The username for the limited user.

- **--limitedpassword**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--limitedpassword</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

Password for the **--limitedpassword**.

- **--optionfile <optionfile>**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--optionfile</td>
</tr>
</tbody>
</table>

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

- **--unattendedmodeui**

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

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

- **--version**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--version</td>
</tr>
</tbody>
</table>

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

- **--agent_installtype**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--agent-installtype</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>database</td>
</tr>
</tbody>
</table>

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

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

**Note**

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

- **--mysqlconnectiongroup**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--mysqlconnectiongroup</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-Line Format</td>
<td>--agentservicename</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value (Windows)</td>
<td>MySQL Enterprise Monitor Agent</td>
</tr>
<tr>
<td>Default Value (Unix)</td>
<td>mysql-monitor-agent</td>
</tr>
<tr>
<td>Default Value (OS X)</td>
<td>mysql.monitor.agent</td>
</tr>
<tr>
<td>Default Value (Linux)</td>
<td>mysql-monitor-agent</td>
</tr>
</tbody>
</table>

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

\begin{itemize}
\item \texttt{--help}
\end{itemize}

\begin{table}[h]
\begin{tabular}{|l|l|}
\hline
\textbf{Property} & \textbf{Value} \\
\hline
Command-Line Format & \texttt{--help} \\
\hline
\end{tabular}
\end{table}

Display the list of valid options to the installer.
Chapter 9 Performance Tuning MySQL Enterprise Monitor

Table of Contents

9.1 Tuning Memory .......................................................................................................................... 79
9.2 Tuning CPU ............................................................................................................................... 81
9.3 Tuning Apache Tomcat Threads ............................................................................................ 81
9.4 Tuning Agent Memory Requirements .................................................................................. 82

There are two major components of the Service Manager that require tuning, the MySQL Instance that is used for the Repository, and the Apache Tomcat application server that serves the Web UI and performs the back-end collection and analysis of data.

9.1 Tuning Memory

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

Tuning Tomcat

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

Table 9.1 Apache Tomcat Environment File Location (default)

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

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

Table 9.2 Installation Parameters

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

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

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

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

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

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

**Important**

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

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

**UNIX, Linux, and Mac**

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

**Microsoft Windows**

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

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

**Tuning InnoDB Memory**

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

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

**Table 9.3 MySQL Enterprise Service Manager repository configuration file location (default)**

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

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

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

To do this, you should:

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

The configuration should be updated with the configuration tool. This tool can be found at the following location:

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

For more information on the configuration utilities, see Chapter 10, Configuration Utilities.

9.3 Tuning Apache Tomcat Threads

When monitoring with a large number of Agent processes deployed, the default number of threads that are created within the Apache Tomcat server may not be sufficient. By default, it is configured to create 150 threads to communicate with the HTTPS port.

This is configured with the maxThreads setting within the server.xml configuration file:

```
<Connector port="18443"
   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
   ...
Tuning Agent Memory Requirements

A good baseline to test would be the number of Agents that you have checking in to the Service Manager plus 50. For example if you have 150 Agents checking in, set the `maxThreads` variable to 200.

Note
* We list default paths to configuration files and tools, so adjust according to where the Service Manager was installed on your system.

9.4 Tuning Agent Memory Requirements

The following are the recommended settings for MySQL Enterprise Monitor Agent:

- A single agent, with default settings and all advisors enabled, should monitor no more than 10 MySQL instances.

- If the agent is monitoring more than 10 MySQL instances, the agent 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 10 Configuration Utilities

Table of Contents

10.1 Service Manager Configuration Utilities ................................................................. 83
10.2 Agent Configuration Utility ...................................................................................... 86

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

10.1 Service Manager Configuration Utilities

Note

The parameters listed here, with the exception of the four repository connection parameters, correspond to those displayed on the Welcome to MySQL Enterprise Monitor page used for initial setup. For more information, see Section 14.1, “Initial Log-In”.

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:

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

Use --help to view the options.

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

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

Service Manager Configuration Utilities

The config script enables you to define or change any of the system configuration parameters such as credentials used to connect to the repository, proxy connection details, and MySQL Enterprise Service Manager user credentials.
Table 10.2 Service Manager Configuration Utilities

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>--mysql-user=&lt;value&gt;</td>
<td>MySQL username for the Service Manager repository. The password is requested via STDIN when the command is run. The default value is service_manager.</td>
<td></td>
</tr>
<tr>
<td>--mu=&lt;value&gt;</td>
<td>MySQL port for the Service Manager repository. The port the target MySQL server listens on. The default is 13306. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--mysql-port=&lt;value&gt;</td>
<td>MySQL database for the Service Manager repository. The name of the database used for the repository. The default is mem. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--mp=&lt;value&gt;</td>
<td>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. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--mysql-db=&lt;value&gt;</td>
<td>Service Manager manager username. The user defined here is added to the manager role.</td>
<td></td>
</tr>
<tr>
<td>--md=&lt;value&gt;</td>
<td>Service Manager agent username. The user defined here is added to the agent role.</td>
<td></td>
</tr>
<tr>
<td>--mysql-server=&lt;value&gt;</td>
<td>Enable automatic checking for online updates. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--ms=&lt;value&gt;</td>
<td>Defines the Query Analyzer data retention policy. Query Analyzer data older than the number of days defined here is deleted. Default is 28 days. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--sm-admin-user=&lt;value&gt;</td>
<td>Defines the historical data retention policy. Historical data older than the number of days defined here is deleted. Default is 28 days. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--auto-update</td>
<td>HTTP Proxy host. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--proxy-host=&lt;value&gt;</td>
<td>HTTP Proxy port. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--proxy-port=&lt;value&gt;</td>
<td>HTTP Proxy username. Requires a server restart if changed.</td>
<td></td>
</tr>
<tr>
<td>--proxy-user=&lt;value&gt;</td>
<td>Checks your repository to ensure compatibility with MySQL Enterprise Service Manager.</td>
<td></td>
</tr>
</tbody>
</table>

**Important**

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

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

```bash
cfg.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:

```bash
cfg.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 10.3 Service Manager Certificate Utilities

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--tomcat-backup-path=&lt;value&gt;</td>
<td>Tomcat backup path to be used to perform upgrade</td>
</tr>
<tr>
<td>--tbp=&lt;value&gt;</td>
<td></td>
</tr>
<tr>
<td>--upgrade</td>
<td>Upgrades certificates on non-OS X platforms. That is, Microsoft Windows, Linux, and so on.</td>
</tr>
<tr>
<td>--upg</td>
<td></td>
</tr>
<tr>
<td>--upgrade-osx</td>
<td>Upgrades certificates on OS X platforms.</td>
</tr>
<tr>
<td>--upgo</td>
<td></td>
</tr>
<tr>
<td>--new-install</td>
<td>Generates a keystore with a new self-signed certificate for a new installation</td>
</tr>
<tr>
<td>--ni</td>
<td></td>
</tr>
<tr>
<td>--accept-keystore-password</td>
<td>If specified, the user is prompted to enter the keystore password. If not specified, the default password changeit is used to access the keystore.</td>
</tr>
<tr>
<td>--akp</td>
<td></td>
</tr>
<tr>
<td>--renew</td>
<td>Renew an existing, self-signed certificate. If the certificate is not self-signed, an error is returned.</td>
</tr>
<tr>
<td>--import-certificate=&lt;value&gt;</td>
<td>Imports the specified certificate. For example:</td>
</tr>
<tr>
<td></td>
<td>--import-certificate=/path/to/client.crt</td>
</tr>
<tr>
<td>--import-key=&lt;value&gt;</td>
<td>Imports the specified private key. For example:</td>
</tr>
</tbody>
</table>
10.2 Agent Configuration Utility

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.

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

Table 10.4 Default Location of Agent Configuration Utility

<table>
<thead>
<tr>
<th>Operating System</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Agent\bin\agent.bat</td>
</tr>
<tr>
<td>UNIX/Linux</td>
<td>/opt/mysql/enterprise/agent/bin/agent.sh</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/agent/bin/agent.sh</td>
</tr>
</tbody>
</table>

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 the MySQL Enterprise Monitor Agent to the monitored MySQL instance.

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

Table 10.5 Agent Connection Utility

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--test-credentials</code></td>
<td>Test MySQL connection credentials.</td>
</tr>
<tr>
<td><code>-T</code></td>
<td></td>
</tr>
<tr>
<td><code>--test-privileges</code></td>
<td>Test admin user's privileges to manage other users.</td>
</tr>
<tr>
<td><code>--create-connection</code></td>
<td>Create or Modify a MySQL connection.</td>
</tr>
<tr>
<td><code>-c</code></td>
<td></td>
</tr>
<tr>
<td><code>--delete-connection</code></td>
<td>Close and Delete a MySQL connection (must also specify --connection-id)</td>
</tr>
<tr>
<td><code>-d</code></td>
<td></td>
</tr>
<tr>
<td><code>--show</code></td>
<td>Show information about all MySQL connections on this agent</td>
</tr>
<tr>
<td><code>-s</code></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--auto-manage-extra-users</code></td>
<td>Auto-create general / limited users (Actions: Create, Modify)</td>
</tr>
<tr>
<td><code>-m</code></td>
<td></td>
</tr>
<tr>
<td><code>--host=&lt;value&gt;</code></td>
<td>Host for the MySQL instance (Actions: Create, Modify)</td>
</tr>
<tr>
<td><code>-h &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--port=&lt;value&gt;</code></td>
<td>Port for the MySQL instance (Actions: Create, Modify)</td>
</tr>
<tr>
<td><code>-P &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--socket=&lt;value&gt;</code></td>
<td>Socket for the MySQL instance (Actions: Create, Modify)</td>
</tr>
<tr>
<td><code>-S &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--limited-user=&lt;value&gt;</code></td>
<td>Limited level credentials (Actions: Create, Modify)</td>
</tr>
<tr>
<td><code>-l &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--general-user=&lt;value&gt;</code></td>
<td>General user credentials</td>
</tr>
<tr>
<td><code>-k &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--admin-user=&lt;value&gt;</code></td>
<td>Admin user credentials</td>
</tr>
<tr>
<td><code>-j &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--connection-id=&lt;value&gt;</code></td>
<td>Connection ID</td>
</tr>
<tr>
<td><code>-i &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--connection-group=&lt;value&gt;</code></td>
<td>MEM Group to use for created/modified connection</td>
</tr>
<tr>
<td><code>-g &lt;value&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>--force-plain-stdin</code></td>
<td>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)</td>
</tr>
<tr>
<td><code>-f</code></td>
<td></td>
</tr>
<tr>
<td><code>--disable-topology-discovery</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>--mysql-identity-source=&lt;value&gt;</code></td>
<td>Source of identity for the MySQL instance for this connection, default or host_plus_datadir. default uses either the server_uuid variable, if present, or generates a new uuid. host_and_datadir uses a hash of the host identity and the path to the MySQL instance's data directory to create a unique identity.</td>
</tr>
<tr>
<td><code>--require-encryption</code></td>
<td>Require the use of TLS for the MySQL connection.</td>
</tr>
<tr>
<td><code>--allow-self-signed-certs</code></td>
<td>When using --require-encryption, allow self-signed TLS certificates.</td>
</tr>
<tr>
<td><code>--ca-file-path=&lt;value&gt;</code></td>
<td>When using --require-encryption, but using a private certificate authority, the path to the CA file.</td>
</tr>
</tbody>
</table>

The following example tests credentials for the root user on localhost:3306:
The following example creates a connection using only the admin user for localhost:3306:

```
agent.bat --test-credentials --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 the MySQL Enterprise Monitor Agent to MySQL Enterprise Service Manager.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--agent-user=&lt;value&gt;</td>
<td>Set the credentials that the Agent uses to connect to the Service Manager</td>
</tr>
<tr>
<td>-u &lt;value&gt;</td>
<td></td>
</tr>
<tr>
<td>--url=&lt;value&gt;</td>
<td>Set the URL for the Service Manager</td>
</tr>
<tr>
<td>-U &lt;value&gt;</td>
<td></td>
</tr>
<tr>
<td>--uuid=&lt;value&gt;</td>
<td>Set the Agent UUID</td>
</tr>
<tr>
<td>-I &lt;value&gt;</td>
<td></td>
</tr>
<tr>
<td>--agent-group=&lt;value&gt;</td>
<td>Set the MEM Group to use for all MySQL connections from this Agent</td>
</tr>
<tr>
<td>-G &lt;value&gt;</td>
<td></td>
</tr>
<tr>
<td>--force-plain-stdin</td>
<td>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)</td>
</tr>
<tr>
<td>-f</td>
<td></td>
</tr>
<tr>
<td>--run-collection-tests</td>
<td>Discover, and attempt to collect OS related assets and dump them to STDOUT (for debugging)</td>
</tr>
<tr>
<td>-t</td>
<td></td>
</tr>
</tbody>
</table>

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

```
agent.sh --agent-user=agent --url=https://localhost:8443
```
Chapter 11 Proxy and Aggregator Installation

Table of Contents

11.1 Prerequisites .................................................................................................................................................. 90
11.2 Installing the Proxy and Aggregator ........................................................................................................... 90
11.3 Graphical Installation Wizard ...................................................................................................................... 91
11.4 Text-Based Installation ................................................................................................................................ 93
11.5 Unattended Installation .................................................................................................................................. 93
11.6 Starting and Stopping the Proxy and Aggregator ......................................................................................... 96
11.7 Configuration Options ...................................................................................................................................... 97

This chapter describes the Proxy, Aggregator and Connector installations and the installation process for the Proxy and Aggregator components.

The MySQL Enterprise Monitor Aggregator requires a framework, or chassis, to handle the communications between the client application and the MySQL instance. The following frameworks are available:

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

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

The following diagram shows a typical MySQL Enterprise Monitor Proxy and Aggregator architecture. In this architecture, the MySQL Enterprise Monitor Proxy and Aggregator installed on the database host. The client application's communication with the database is intercepted by the proxy.

Figure 11.1 MySQL Enterprise Monitor Proxy and Aggregator Architecture
Prerequisites

Note
The MySQL Enterprise Monitor Proxy and Aggregator does not have to be installed on the same host as the monitored MySQL instance. You can install it on another host.

Important
The MySQL Enterprise Monitor Aggregator is supported by the PHP Connector. The other Connectors do not require the Aggregator and can communicate directly with the MySQL Enterprise Service Manager once configured to do so. For more information on configuring the Connectors, see Chapter 12, Configuring Connectors.

11.1 Prerequisites

This section describes the installation prerequisites.

Proxy and Aggregator Users

It is not recommended to install the MySQL Enterprise Monitor Proxy and Aggregator as root. It is recommended to create a user specifically for the Proxy and Aggregator and install the products as that user, usually the mysql user. The same is true for the MySQL Enterprise Monitor Aggregator installation.

For MySQL 5.5 and earlier, the MySQL client-application user must have SELECT privileges on the mysql.inventory table. This table contains the server UUID which is required to report the Query Analyzer data to the MySQL Enterprise Service Manager. Use the GRANT statement. For example:

```
mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
```

Note
This privilege is not required for MySQL 5.6 or higher, as the mysql.inventory table is not created.

Performance Schema

If you are using the MySQL Enterprise Monitor Proxy and Aggregator to collect query performance data, you must ensure the statements_digest consumer in performance_schema.setup_consumers is disabled.

11.2 Installing the Proxy and Aggregator

The following installations are possible:

• Aggregator and Proxy: Proxy and Aggregator are installed and configured together.

• Aggregator: Aggregator is installed without the Proxy. Only use this installation type if you intend to use the Aggregator with MySQL Connector/PHP.

The installer has the following filename convention:

```
mysqlmonitoraggregator-version_number-platform-architecture-installer.extension
```

where:
• *version_number* is the version number of the product.

• *platform* is the intended operating system for the installer.

• *architecture* specified whether the installer is for 32- or 64-bit platforms. If no architecture is present, the installer is 32-bit.

The installers support the following installation types:

• Graphical Installation Wizard

• Text mode

• Unattended mode

### 11.3 Graphical Installation Wizard

This section describes how to install the MySQL Enterprise Monitor Proxy and Aggregator together using the MySQL Enterprise Monitor Proxy and Aggregator Installation Wizard. This process is identical across all platforms, except where explicitly stated.

This installation package installs one of the following:

• **Proxy and Aggregator**: installs both the Proxy and Aggregator.

• **Aggregator Only**: installs the Aggregator only.

To install the MySQL Enterprise Monitor Proxy and Aggregator using the Graphical Installation Wizard, do the following:

**Note**

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

**Starting the Installation**

1. Run the installer as required by the operating system.

   The language selection dialog is displayed. Choose a language and click **Next**.

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

   The **Installation Directory** dialog is displayed.

3. Define an installation directory, or accept the default installation directory, and click **Forward**.

   The component selection page is displayed.

   If you choose **Proxy and Aggregator**, you must follow the steps in Installing the Proxy and Installing the Aggregator

   If you choose **Aggregator Only**, you must follow the steps in Installing the Aggregator.

**Important**

There is no difference in the files installed. The **Aggregator Only** option installs all files, Proxy included, but only configures the Aggregator. The Proxy files
Installing the Proxy

This section describes how to install the MySQL Enterprise Monitor Proxy. To install the MySQL Enterprise Monitor Proxy, do the following:

1. Enter the port number the Proxy uses to listen for incoming connections. The default port is 6446.

2. Select the communication protocol the Proxy uses to connect to the monitored MySQL instance. Note Socket is not available on Windows platforms.

If you intend to use socket to connect to the database, select Socket and click Forward to define the path to the socket you want to use. After the socket is defined, click Forward to proceed with the installation.

If you intend to use TCP/IP, select TCP/IP and click Forward to proceed with the installation.

The MySQL database configuration dialog is displayed.

3. Enter the hostname or IP address and the port number of the monitored MySQL instance.

Click Forward.

The Aggregator configuration dialog is displayed.

4. The Proxy installation configuration is completed. The MySQL Enterprise Monitor Aggregator installation configuration is described in Installing the Aggregator.

Installing the Aggregator

This section describes how to install the MySQL Enterprise Monitor Aggregator. To install the MySQL Enterprise Monitor Aggregator, do the following:

1. Complete the following fields on the Aggregator configuration dialog:
   - Aggregator Port: the port the Aggregator listens on.
   - Aggregator SSL Port: the port the Aggregator listens on for SSL communication.
   - PEM Certificate file: the location of the PEM certificate.
   - CA Certificate file: the location of the CA certificate.

Click Forward to continue.

The MySQL Enterprise Monitor options dialog is displayed.

2. Complete the MySQL Enterprise Monitor option fields. This information is used by the Aggregator to connect to the MySQL Enterprise Service Manager.
You must provide the following information:

- **Hostname or IP address**: the address of the MySQL Enterprise Service Manager installation.
- **Tomcat SSL Port**: the port Tomcat is listening on for SSL connections.
- **Agent Username**: the username of the Agent. These are the connection credentials the Aggregator uses to connect to the MySQL Enterprise Service Manager.
- **Agent Password**: the password of the Agent.

Click **Forward**. The **Configuration Report** is displayed.

3. Review the data in the **Configuration Report** to ensure all configuration settings are correct.

Click **Forward** to complete the installation.

### 11.4 Text-Based Installation

The steps and options of the text-based installation are identical to those described in Section 11.3, "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:

   ```shell
text-mode
   ```

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

   ```shell>
   ./mysqlmonitoraggregator-3.0.14.3040-linux-x86-64bit-installer.bin --mode text
   ```

   The text installation process starts.

2. Follow the instructions onscreen. The options and values are identical to those described in Section 11.3, "Graphical Installation Wizard".

### 11.5 Unattended Installation

The MySQL Enterprise Monitor Proxy and Aggregator installers enable you to perform unattended installations. This is useful for large scale installations on multiple machines. The installations can be run using all required options on a command line, or by defining the required options in a configuration file and calling that file for each installation.

**Unattended Installation Options**

To display the installation options available, run the installer from the command line with the following option:

```shell
--help
```

The following options are available:
### Table 11.1 MySQL Enterprise Monitor Proxy and Aggregator Installer Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>--help</strong></td>
<td>Displays the help text listing all options available for the platform on which the installer was run.</td>
</tr>
<tr>
<td></td>
<td>On Microsoft Windows platforms, this option does not output the list of options in the console window, but in a separate help window.</td>
</tr>
<tr>
<td><strong>--version</strong></td>
<td>Displays the product version.</td>
</tr>
<tr>
<td></td>
<td>On Microsoft Windows platforms, this option does not output the version details in the console window, but in a separate help window.</td>
</tr>
<tr>
<td><strong>--debuglevel</strong></td>
<td>Sets the verbosity of the installation log. 0 is the lowest verbosity, 4 is the highest. Default value is 2.</td>
</tr>
<tr>
<td><strong>--debugtrace</strong></td>
<td>Sets the path and filename of the installation log file.</td>
</tr>
<tr>
<td><strong>--optionfile</strong></td>
<td>Sets the path and filename of the installation options file. For more information, see Unattended Installation with Options File.</td>
</tr>
<tr>
<td><strong>--installer-language</strong></td>
<td>Sets the language of the installation. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• en: English. Default value.</td>
</tr>
<tr>
<td></td>
<td>• ja: Japanese.</td>
</tr>
<tr>
<td><strong>--mode</strong></td>
<td>Sets the installation mode. This varies according to the platform. For example, on Linux-based systems, you can choose a GUI-based installer with <strong>--mode gtk</strong>, or choose a text-only, console-based installation with <strong>--mode text</strong>.</td>
</tr>
<tr>
<td></td>
<td>The following is a list of the GUI-based installation options available:</td>
</tr>
<tr>
<td></td>
<td>• Windows: <strong>Win32</strong></td>
</tr>
<tr>
<td></td>
<td>• OS X: <strong>osx</strong></td>
</tr>
<tr>
<td></td>
<td>• Solaris: <strong>xwindow</strong></td>
</tr>
<tr>
<td></td>
<td>• Linux: <strong>gtk</strong> (Default) and <strong>xwindow</strong></td>
</tr>
<tr>
<td><strong>--mode</strong></td>
<td>Can also initiate text mode and unattended installations.</td>
</tr>
<tr>
<td><strong>--mode text</strong></td>
<td>Starts a text-only, console-based installation process. Text-based installation are not available on Windows platforms.</td>
</tr>
<tr>
<td><strong>--mode unattended</strong></td>
<td>Starts an unattended installation. For more information on unattended installations, see Unattended Installation from the Command Line and Unattended Installation with Options File.</td>
</tr>
</tbody>
</table>
## Unattended Installation Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--unattendedmodeui</td>
<td>Sets the graphical elements to use, if any, in the unattended installation. The following options are available:</td>
</tr>
<tr>
<td></td>
<td>• none: no pop-ups, or progress bars are displayed. Errors are displayed, if they occur.</td>
</tr>
<tr>
<td></td>
<td>• minimal: No user interaction is required and a progress bar is displayed showing the installation progress. Errors are displayed, if they occur.</td>
</tr>
<tr>
<td></td>
<td>• minimalWithDialogs:</td>
</tr>
<tr>
<td>--installdir</td>
<td>Sets the installation directory for the product.</td>
</tr>
<tr>
<td>--use-external-glib</td>
<td>Sets the glib to use, the one delivered in the installer (0, default), or the system glib (1).</td>
</tr>
<tr>
<td>--monitorcomponent</td>
<td>Specifies which component to install. The following options are available:</td>
</tr>
<tr>
<td></td>
<td>• proxy: installs both Proxy and Aggregator. This is the default.</td>
</tr>
<tr>
<td></td>
<td>• aggregator: installs the Aggregator only.</td>
</tr>
</tbody>
</table>

### Proxy-Specific Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxyservicename</td>
<td>Sets the unique service name for the Proxy service.</td>
</tr>
<tr>
<td>--mysqlconnmethod</td>
<td>Sets the connection method used by the proxy to connect to the monitored MySQL instance. The following options are available:</td>
</tr>
<tr>
<td></td>
<td>• tcpip: default value.</td>
</tr>
<tr>
<td></td>
<td>• socket: unavailable on Microsoft Windows platforms.</td>
</tr>
<tr>
<td>--proxyport</td>
<td>Sets the port the Proxy listens on for incoming connections. Default value is 6446.</td>
</tr>
<tr>
<td>--mysqlhost</td>
<td>Sets the hostname or IP address of the monitored MySQL instance. Default value is localhost.</td>
</tr>
<tr>
<td>--mysqlport</td>
<td>Sets the port of the monitored MySQL instance. Default value is 3306.</td>
</tr>
<tr>
<td>--mysqlsocket</td>
<td>Sets the socket used by the monitored MySQL instance.</td>
</tr>
</tbody>
</table>

### Aggregator-specific Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--aggregatorservicename</td>
<td>Sets the unique name for the Aggregator service. Default value is mysql-monitor-aggregator.</td>
</tr>
<tr>
<td>--aggregatorport</td>
<td>Sets the port the Aggregator listens on. Default value is 14000.</td>
</tr>
<tr>
<td>--aggregatorsslport</td>
<td>Sets the SSL port the Aggregator listens on for secure connections. Default value is 14443.</td>
</tr>
<tr>
<td>--aggregatorsslcertfile</td>
<td>Sets the location of the SSL certificate.</td>
</tr>
<tr>
<td>--aggregatorsslcafile</td>
<td>Sets the location of the SSL CA file.</td>
</tr>
<tr>
<td>--managerhost</td>
<td>Sets the hostname or IP address of the MySQL Enterprise Service Manager installation. Default value is localhost.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--managerport</td>
<td>Sets the SSL port number of the MySQL Enterprise Service Manager's Tomcat installation. Default value is 18443.</td>
</tr>
<tr>
<td>--agentuser</td>
<td>Sets the agent username which the Aggregator uses to communicate with the MySQL Enterprise Service Manager. Default value is agent.</td>
</tr>
<tr>
<td>--agentpassword</td>
<td>Sets the password of the agent used by the Aggregator.</td>
</tr>
</tbody>
</table>

### Unattended Installation from the Command Line

To run the unattended installation from the command line, enter the installer name, followed by the --mode unattended option, followed by the options you want to define. If you do not define an option on the command line, the default value is used, if a default exists. If no default value exists, you must define that value in the configuration after the installation is complete.

The following example installs MySQL Enterprise Monitor Proxy and Aggregator on a Linux platform but changes the MySQL Enterprise Service Manager values:

```
./mysqlmonitoraggregator-4.0.3.3041-linux-x86-b4bit-installer.bin --mode unattended
--unattendedmodeui none --managerhost service.manager.com --agentuser Agent100
--agentpassword D4unKotR
```

This example changes the following:

- Instructs the installer to display no dialogs of any kind. In this mode, errors are displayed if they occur.
- Sets the MySQL Enterprise Service Manager location to service.manager.com. This is the location of your MySQL Enterprise Service Manager installation. The default ports were not changed.
- Sets the Agent username to Agent100 and the Agent password to D4unKotR.

### Unattended Installation with Options File

If you use an options file, add the options you want to change to a text file as name=value pairs. Using the example shown in Unattended Installation from the Command Line, the text file contents are:

```
mode=unattended
unattendedmodeui=none
managerhost=service.manager.com
agentuser=Agent100
agentpassword=D4unKotR
```

If this file was saved as pa-options.txt, the installation command takes the following format:

```
./mysqlmonitoraggregator-3.4.5.4541-linux-x86-b4bit-installer.bin --optionfile pa-options.txt
```

### 11.6 Starting and Stopping the Proxy and Aggregator

This section describes how to start and stop the MySQL Enterprise Monitor Proxy and Aggregator.

On UNIX, Linux, and Mac OS X platforms, the Proxy and Aggregator processes are controlled using the scripts in the etc/init.d directory of your installation. On Windows platforms, you can start, stop and restart your services using the Start menu entries, or through the Services control of the Microsoft Management Console.
Important

If you install the MySQL Enterprise Monitor Proxy and Aggregator, both Proxy and Aggregator run under the name of the MySQL Enterprise Monitor Proxy, not as two distinct services. If you install the MySQL Enterprise Monitor Aggregator standalone, it is run as the MySQL Enterprise Monitor Aggregator.

- **init.d:**
  - Starting the Proxy and Aggregator: run `./mysql-monitor-proxy start`.
  - Stopping the Proxy and Aggregator: run `./mysql-monitor-proxy stop`.
  - Starting the Aggregator: run `./mysql-monitor-aggregator start`.
    
    If you have installed both Proxy and Aggregator, do not run this command. The Aggregator is started by the Proxy-specific commands.
  - Stopping the Aggregator: run `./mysql-monitor-aggregator stop`.
    
    If you have installed both Proxy and Aggregator, do not run this command. The Aggregator is stopped by the Proxy-specific commands.
  - Status: run either script with the `status` option to see the status of the service.
    
    If you installed both Proxy and Aggregator, the status returns information on the Proxy only. In this installation type, if the Proxy is running, the Aggregator is running also. For more information, check the `mysql-monitor-proxy.log`.
  - Restarting: run either script, depending on your installation type, with the `restart` option to restart the services.

11.7 Configuration Options

It is possible to run the Proxy, or Aggregator, or both, with specific options, using the following files installed in the `bin` directory of your installation:

- `mysql-monitor-aggregator`
- `mysql-monitor-proxy`

Note

On Windows platforms, these files are executables and have the `exe` extension. On Linux, UNIX and Mac platforms, they are shell scripts.

To view the options available, run either file with the `--help` option.

The help output is broken down into the following sections:

- Help Options: lists the various help output options.
- Application Options: lists the application options.
- aggr-module: lists the Aggregator-specific options. Displayed only for the `--help-all` option.
- proxy-module: lists the Proxy-specific options. Displayed only for the `--help-all` option.
The `mysql-monitor-aggregator` help displays the application and Proxy module help, only. The `mysql-monitor-proxy` help displays application, aggregator and proxy output.

The following table lists the Proxy and Aggregator help options:

### Table 11.2 Proxy and Aggregator Help Options

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

The following table lists the Proxy and Aggregator command line options:

### Table 11.3 Proxy and Aggregator Command Line Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-V, --version</td>
<td>Shows the version of the Proxy or Aggregator, depending on which file it is run with.</td>
</tr>
<tr>
<td>--defaults-file=&lt;file&gt;</td>
<td>Defines a configuration file to use. Similarly to running an unattended installation with an options file, this enables you to define all configuration changes as name-value pairs (without the <code>--</code> prefix for each option) and call the file as needed.</td>
</tr>
<tr>
<td>--verbose-shutdown</td>
<td>Configures the application to always log the exit code on shutdown.</td>
</tr>
<tr>
<td>--daemon</td>
<td>Configures the application to run in daemon mode.</td>
</tr>
<tr>
<td>--user=&lt;user&gt;</td>
<td>Defines the specific user to run the Aggregator.</td>
</tr>
<tr>
<td>--basedir=&lt;absolute path&gt;</td>
<td>Defines the absolute path of the base directory which is prefixed to all relative paths in the configuration. If you define a relative path, an error is returned.</td>
</tr>
<tr>
<td>--pid-file=&lt;file&gt;</td>
<td>Defines the name of the PID file to use in the event the application is started in daemon mode.</td>
</tr>
<tr>
<td>--plugin-dir=&lt;path&gt;</td>
<td>Defines the path to the plugins.</td>
</tr>
<tr>
<td>--plugins-name=&lt;name&gt;</td>
<td>Defines the names of the plugins to load. On the command line, you can specify this value multiple time. In the configuration file, the option is entered once, followed by a comma-separated list of the required plugins.</td>
</tr>
<tr>
<td>--log-level=&lt;string&gt;</td>
<td>Defines the logging level. Possible values are <code>critical</code> (default value), <code>error</code>, <code>warning</code>, <code>info</code>, <code>message</code>, and <code>debug</code>.</td>
</tr>
<tr>
<td>--log-file=&lt;filename&gt;</td>
<td>Defines the name of the logfile.</td>
</tr>
<tr>
<td>--log-use-syslog</td>
<td>Configures the application to send all messages to the syslog. UNIX/Linux only.</td>
</tr>
<tr>
<td>--log-backtrace-on-crash</td>
<td>Configures the application to invoke the debugger in the event of a crash.</td>
</tr>
<tr>
<td>--keepalive</td>
<td>Configures the application to attempt a restart in the event of a crash. Not available on Microsoft Windows. When running as a service, the Proxy automatically restarts.</td>
</tr>
<tr>
<td>--max-open-files</td>
<td>Defines the maximum number of open files.</td>
</tr>
</tbody>
</table>
The following table lists the Aggregator module command line options:

### Table 11.4 aggr-module Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--event-threads</td>
<td>Configures the number of event-handling threads. Default value is 1.</td>
</tr>
<tr>
<td>--lua-path=&lt;path&gt;</td>
<td>Sets the LUA_PATH.</td>
</tr>
<tr>
<td>--lua-cpath=&lt;path&gt;</td>
<td>Sets the LUA_CPATH.</td>
</tr>
<tr>
<td>--aggr-address=<a href="">host:port</a></td>
<td>Defines the address and listening port of the Aggregator. The default port value is 14000.</td>
</tr>
<tr>
<td>--aggr-lua-script=&lt;filename&gt;</td>
<td>Defines the path to the LUA script.</td>
</tr>
<tr>
<td>--aggr-mem-url=&lt;url&gt;</td>
<td>Defines the URL to the MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td>--aggr-mem-user=&lt;string&gt;</td>
<td>Defines the Agent username to use for communication with the MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td>--aggr-mem-password=&lt;string&gt;</td>
<td>Defines the Agent password to use for communication with the MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td>--aggr-ssl-address=<a href="">host:port</a></td>
<td>Defines the address and listening port of the Aggregator for SSL connections to the Aggregator.</td>
</tr>
<tr>
<td>--aggr-ssl-cert-file=&lt;filename&gt;</td>
<td>Defines the PEM server certificate for the Aggregator.</td>
</tr>
<tr>
<td>--aggr-ssl-cs-file=&lt;filename&gt;</td>
<td>Defines the CA certificate for the Aggregator.</td>
</tr>
<tr>
<td>--aggr-ssl-ciphers=&lt;string&gt;</td>
<td>Defines the supported ciphers.</td>
</tr>
<tr>
<td>--aggr-test-mode</td>
<td>Start the Aggregator in test mode. This mode ignores the flush interval setting and aggregates queries until instructed to return the aggregated data by a HTTP REST interface. It returns a JSON result set of all the normalized queries and their aggregated data.</td>
</tr>
<tr>
<td>--aggr-flush-interval=&lt;seconds&gt;</td>
<td>Defines the interval, in seconds, at which the query data is flushed to the MySQL Enterprise Service Manager. The default value is 60 seconds.</td>
</tr>
<tr>
<td>--aggr-max-request-body-size=&lt;bytes&gt;</td>
<td>Defines the maximum size of an HTTP request body. The default size is 1MB.</td>
</tr>
</tbody>
</table>

The following table lists the Proxy module command line options:

### Table 11.5 proxy-module Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-P, --proxy-address=<a href="">host:port</a></td>
<td>The address and listening port of the Proxy. Default port is 4040.</td>
</tr>
<tr>
<td>-r, --proxy-read-only-backend-addresses</td>
<td>The address and listening port of the remote, slave server. This is not set by default.</td>
</tr>
<tr>
<td>-b, --proxy-backend-addresses=<a href="">host:port</a></td>
<td>The host name (or IP address) and port of the MySQL server to connect to. You can specify multiple backend servers by supplying multiple</td>
</tr>
</tbody>
</table>
### Configuration Options

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>options. Clients are connected to each backend server in round-robin fashion.</td>
</tr>
<tr>
<td></td>
<td>For example, if you specify two servers A and B, the first client connection goes to server A; the second client connection to server B and the third client connection to server A. When using this option on the command line, you can specify the option and the server multiple times to specify multiple backends.</td>
</tr>
<tr>
<td></td>
<td>When using this option in a configuration file, separate multiple servers with commas.</td>
</tr>
<tr>
<td>--proxy-skip-profiling</td>
<td>Disable query profiling (statistics time tracking). The default is for tracking to be enabled.</td>
</tr>
<tr>
<td>-s file_name,--proxy-lua-script=&lt;file&gt;</td>
<td>The Lua script file to be loaded. The script file is not loaded and parsed until a connection is made. Also note that the specified Lua script is reloaded for each connection; if the content of the Lua script changes while the Proxy is running, the updated content is automatically used when a new connection is made.</td>
</tr>
<tr>
<td>--no-proxy</td>
<td>Disables the Proxy module. By default, the Proxy is enabled.</td>
</tr>
<tr>
<td>--proxy-pool-no-change-user</td>
<td>Disable use of the MySQL protocol CHANGE_USER command when reusing a connection from the pool of connections specified by the proxy-backend-addresses list.</td>
</tr>
<tr>
<td>--proxy-connect-timeout</td>
<td>Defines the Proxy's connection timeout in seconds. Default value is 2.</td>
</tr>
<tr>
<td>--proxy-read-timeout</td>
<td>Defines the read timeout in seconds. Default is 8 hours.</td>
</tr>
<tr>
<td>--proxy-write-timeout</td>
<td>Defines the write timeout in seconds. Default is 8 hours.</td>
</tr>
</tbody>
</table>

These options, with the exception of the help, version and defaults-file options, are also used, as name=value pairs, in the ini files used to configure the Proxy and Aggregator services.

The configuration files are located in the etc directory of your installation.

- **mysql-monitor-proxy.ini**: configures the Proxy and Aggregator. Use this file when both components are installed.
- **mysql-monitor-aggregator.ini**: configures the Aggregator. Use this file when only the Aggregator is installed.
Chapter 12 Configuring Connectors

Table of Contents

12.1 Using the MySQL Enterprise Plugin for Connector/PHP ................................................................. 101
12.2 Using the MySQL Enterprise Plugin for Connector/J ................................................................. 105
12.3 Using the MySQL Enterprise Plugin for Connector/.NET ......................................................... 110

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

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

• Section 12.2, “Using the MySQL Enterprise Plugin for Connector/J”: describes the configuration of the Connector/J to feed query information to the Query Analyzer.

• Section 12.3, “Using the MySQL Enterprise Plugin for Connector/.NET”: describes the configuration of the Connector/.NET to feed query information to the Query Analyzer.

12.1 Using the MySQL Enterprise Plugin for Connector/PHP

The MySQL Enterprise Plugin for Connector/PHP enables you to use the Query Analyzer to monitor MySQL queries from PHP applications, such as PHP-enabled web pages. The Query Analyzer enables you to locate and analyze queries that are inefficient or slow. Tuning such queries helps to shorten load times for web pages, and improves overall system responsiveness and scalability.

The PHP query data is routed through the MySQL Enterprise Monitor Aggregator. The Aggregator receives query information from the PHP plug-in, aggregates and computes statistics, and sends this data to the MySQL Enterprise Service Manager, where it is displayed by the Query Analyzer. You must have the MySQL Enterprise Monitor Aggregator enabled and running to use Query Analyzer with PHP applications.

Figure 12.1 Plug-in for PHP and Aggregator Architecture
Prerequisites

The MySQL Enterprise Plugin for Connector/PHP requires PHP 5.3.2 or above, with the MySQL native driver, `mysqlnd`, installed. This is the recommended configuration. If your PHP installation was not configured with the `mysqlnd` enabled, you must rebuild and install PHP from source using at least one of the following options:

- `--with-mysqli=mysqlnd`
- `--with-pdo-mysql=mysqlnd`
- `--with-mysql=mysqlnd`

The preceding options are supplied to the `configure` command, depending on which extension you are using (`mysql`, `mysqli` or `PDO_MYSQL`). If you use more than one extension, provide multiple options. Specifying any of the options listed rebuilds PHP with `mysqlnd` support. You also must enable the PHP JSON module.

For MySQL 5.5, or earlier, the MySQL client application user, must have `SELECT` privileges on the `mysql.inventory` table. This table contains the server UUID required to report the Query Analyzer data to the MySQL Enterprise Service Manager. Use the `GRANT` statement. For example:

```sql
mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
```

Installation

The plug-in is provided as a regular PHP module (PHP extension), and installation follows those PHP standard procedures as described on [http://php.net/install.pecl](http://php.net/install.pecl).

Download the MySQL Enterprise Plugin for Connector/PHP, then use the following step-by-step instructions to install and configure the MySQL Enterprise Plugin for Connector/PHP extension.

1. Locate your `php.ini` configuration file. If you do not know the location, you can view information about your PHP installation by creating a script containing:

   ```php
   <?php phpinfo(); ?>
   ```

   Place the script within a directory configured for providing PHP web pages. Now load the page in your web browser to see a list of configuration and other information about your PHP installation.

   Check the output for **Loaded Configuration File**. If the value is *(none)*, refer to the **Configuration File (php.ini) Path** and create a file called `php.ini` in there. If a **Scan this dir for additional .ini files** option is listed you can also create a file using any name you like, ending `.ini`, in that directory to set configuration options.

2. Identify whether or not your PHP build was built "thread safe" by checking the **Thread Safety** value in the output from the `phpinfo()` test. If your PHP build is thread safe, you need
mysqlenterprise_ts.so on Linux, Unix, and OS X, or php_mysqlenterprise_ts.dll on Microsoft Windows. If not, use mysqlenterprise.so on Linux, Unix, and OS X, or php_mysqlenterprise.dll on Microsoft Windows.

3. Add an entry for the MySQL Enterprise Plugin for Connector/PHP module. The following example uses the full path:

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

Alternatively, add the file to the directory defined by the `extension_dir` configuration option, and specify the file name:

```
extension=mysqlenterprise.so
```

### Note

If `mysqlnd` is loaded as a shared library (`mysqlnd.so`), it must be loaded before `mysqlenterprise.so` or errors such as PHP Warning: PHP Startup: Unable to load dynamic library `/mysqlenterprise.so' - `/mysqlenterprise.so: undefined symbol: mysqlnd_plugin_register in Unknown on line 0` are emitted by PHP. Either:

- If `php.ini` is used to load the PHP extensions, then list it first. For example:

  ```
  extension=mysqlnd.so
  extension=mysqlenterprise.so
  ```

- If individual ini files are used to load the PHP extensions, the ini files are loaded alphabetically, adjust accordingly so that `mysqlnd.so` is loaded first. For example, `/etc/php.d/` might contain:

  ```
  mysqlnd.ini
  mysqlzz_enterprise.ini
  ```

4. Users of Debian-based systems, such as Ubuntu, are encouraged to use the `php5enmod` command to enable extensions. For example:

```
shell> php5enmod /path/to/mysqlenterprise.so
```

`php5enmod` creates a symlink from the usual `conf.d` directory that points to where the real files are located in `mods-available`, and prefixes it with a priority number.

5. Restart your Web server application to reload PHP and the configured extensions.

6. Reload the `phpinfo()` page, and inspect the listing for the `mysqlenterprise` module.

### Caution

If you are using PHP on Microsoft Windows with the Apache web server (httpd) built from apache.org, note the following:

MySQL no longer supports VC6, the MySQL Enterprise Plugin for Connector/PHP for Microsoft Windows is compiled with the newer VC9 compiler. You can not use PHP as a loaded module with an Apache web server build that uses VC6.
Alternative Apache builds exist that use VC9. Check your source and ensure that your binaries are compiled using VC9.

PHP binaries for Microsoft Windows from php.net have compiled mysqlnd support by default, since PHP 5.3.0.

Configuration

The configuration of the MySQL Enterprise Plugin for Connector/PHP is handled through the standard PHP configuration files, either globally using `php.ini`, or by using the per-directory options, as detailed in PHP Configuration. The following table shows the available configurable options.

**Note**

Each PHP configuration option for MySQL Enterprise Monitor is prefixed by `mysqlenterprise`.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| **aggregator_connect_timeout_sec** | Timeout, in seconds, for communications with the MySQL Enterprise Monitor Aggregator.  
  • Property type: integer  
  • Default value: 1  
  This property can be combined with the `aggregator_connect_timeout_usec` property. |
| **aggregator_connect_timeout_usec** | Timeout, in microseconds, for communications with the MySQL Enterprise Monitor Aggregator.  
  • Property type: integer  
  • Default value: 0  
  This property can be combined with the `aggregator_connect_timeout_sec` property. |
| **aggregator_user** | The Aggregator's username. See Chapter 11, Proxy and Aggregator Installation for more information.  
  • Property type: string |
| **aggregator_password** | The Aggregator's password.  
  • Property type: string |
| **aggregator_url** | The IP address, or hostname, and port of the Aggregator installation.  
  • Property type: string  
  • Default value: `tcp://127.0.0.1:14000` |
| **debug_callback** | This property should be used only when debugging your MySQL Enterprise Monitor installation with MySQL Support personnel. |
### Property Description

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defines the name of the callback function to invoke when data is sent to the Aggregator. The callback is defined in the PHP application and is a function which requires a single parameter, the array of HTTP requests made to the Aggregator.</td>
</tr>
<tr>
<td>disable_backtrace</td>
<td>Defines whether a backtrace is performed. Backtrace is useful for debugging but has a performance impact.</td>
</tr>
<tr>
<td></td>
<td>• Property type: boolean</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1, the backtrace is disabled. To enable the backtrace, set this value to 0.</td>
</tr>
<tr>
<td>log_file</td>
<td>Defines the location of a log file which logs all query information sent to the Aggregator.</td>
</tr>
<tr>
<td></td>
<td>This should only be used for debugging purposes because every request is logged, resulting in a very large log file.</td>
</tr>
<tr>
<td></td>
<td>• Property type: string</td>
</tr>
<tr>
<td>quan_enabled</td>
<td>Defines whether query analysis is enabled.</td>
</tr>
<tr>
<td></td>
<td>• Property type: boolean</td>
</tr>
<tr>
<td></td>
<td>• Default value: 1, query analysis is enabled. To disable query analysis, set this value to 0.</td>
</tr>
</tbody>
</table>

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

```ini
extension = /usr/local/apache/php/lib/php/extensions/mysqlenterprise.so
mysqlenterprise.aggregator_url = tcp://aggregator:14000
mysqlenterprise.quan_enabled = 1
mysqlenterprise.debug_callback = cta_callback
mysqlenterprise.disable_backtrace = 1
mysqlenterprise.aggregator_user = username
mysqlenterprise.aggregator_password = "password"
```

**Note**

You must restart your server after setting these properties. Verify the settings are correct by checking the output of `phpinfo()`.

### 12.2 Using the MySQL Enterprise Plugin for Connector/J

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

**Important**

The Connector/J does not require the MySQL Enterprise Monitor Aggregator for query aggregation.
Prerequisites

1. MySQL Connector/J version 5.1.12 or later.
2. JDK-1.7.0 or later.
3. MySQL Enterprise Service Manager version 3.2 or later.
4. For MySQL 5.5, or earlier, the MySQL client application user must have `SELECT` privileges on the `mysql.inventory` table. This table contains the server UUID which is required to report the Query Analyzer data to the MySQL Enterprise Service Manager. Use the `GRANT` statement. For example:

```sql
code
mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
```

5. Apache Commons logging in the `CLASSPATH` of the application being analyzed. If you are not already using Commons Logging, modify the application's `CLASSPATH` as described in the following section to point to the JAR file bundled with the MySQL Enterprise Monitor product.

Installation

Place the JAR file `lib/c-java-mysql-enterprise-plugin-version.jar` in the application's `CLASSPATH` where it is visible to the version of MySQL Connector/J in use. Ideally, use the same location as MySQL Connector/J's JAR file, or in a parent classloader to that JAR file's location.

If the application being analyzed does not have Apache Commons Logging in the `CLASSPATH`, install `lib/required/commons-logging-1.1.1.jar` in the application's `CLASSPATH`. If no other component in your application uses Apache Commons Logging, install it in the same directory where the plug-in is installed.

There is static shutdown() method on `com.mysql.etools.jdbc.StatementPerformanceCounters`, which can be used to cleanly shutdown the query analysis plug-in when the application is going to be shutdown.
If the application is deployed in a J(2)EE application server, there is a `ContextListener` distributed with the plug-in which calls this method when the application’s context is shutdown (or reloaded). Application Servers which support `@WebListener` (such as JEE6 and above) do not need to do any extra configuration, but users with older Application Servers need to add the following line to their application’s `web.xml` file:

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

### Using the MySQL Enterprise Plug-in for Connector/J

This section describes how to configure the MySQL Plug-in for Connector/J.

#### Table 12.2 MySQL Plug-in for Connector/J Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>statementInterceptors</code></td>
<td>Enables the plug-in. Set this property as follows: <code>statementInterceptors = com.mysql.etools.jdbc.StatementPerformanceCounters</code></td>
</tr>
<tr>
<td><code>disableSourceLocation</code></td>
<td>Defines whether to send stack traces with example queries to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>boolean</code></td>
</tr>
<tr>
<td></td>
<td>• Values: <code>true</code> or <code>false</code> (default).</td>
</tr>
<tr>
<td><code>serviceManagerUrl</code></td>
<td>Defines the URL of the MySQL Enterprise Service Manager. Include the full URL and port number.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>string</code></td>
</tr>
<tr>
<td></td>
<td>• Value: URL and port number of MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td><code>serviceManagerUser</code></td>
<td>Defines the Agent username to use when connecting to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>string</code></td>
</tr>
<tr>
<td><code>serviceManagerPassword</code></td>
<td>Defines the Agent password to use when connecting to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>string</code></td>
</tr>
<tr>
<td><code>serviceManagerConnectTimeout</code></td>
<td>Defines the number of seconds to wait for a connection to MySQL Enterprise Service Manager.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <code>numeric</code></td>
</tr>
<tr>
<td></td>
<td>• Default value: 0</td>
</tr>
<tr>
<td><code>serviceManagerResponseTimeout</code></td>
<td>Defines the number of seconds to wait for a response from MySQL Enterprise Service Manager.</td>
</tr>
</tbody>
</table>
Using the MySQL Enterprise Plug-in for Connector/J

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysqlServerUUID</td>
<td>If you are unable to retrieve the server's UUID, define it with this property.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>string</strong></td>
</tr>
<tr>
<td></td>
<td>For MySQL 5.5, or earlier, to retrieve the UUID, the plug-in requires <code>SELECT</code> privileges on <code>mysql.inventory</code>.</td>
</tr>
<tr>
<td>verifySslHostnames</td>
<td>If set to true, it enables verification of the host names in the SSL Server certificate. Host names are verified using the same schema as used by Firefox, and Curl, and specified by RFC 2818.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>boolean</strong></td>
</tr>
<tr>
<td></td>
<td>• Default value: <strong>false</strong></td>
</tr>
<tr>
<td>verifySslCerts</td>
<td>Defines whether the plug-in verifies the certificate presented by the server was signed by a CA in the <code>trustCertificateKeystore</code>.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>boolean</strong></td>
</tr>
<tr>
<td></td>
<td>• Default: <strong>false</strong>, verification is disabled.</td>
</tr>
<tr>
<td>trustCertificateKeystoreUrl</td>
<td>Defines the URL of the trusted root certificate KeyStore. If none is specified, the Java defaults are used.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>string</strong></td>
</tr>
<tr>
<td>trustCertificateKeystorePassword</td>
<td>Defines the password for the KeyStore.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>string</strong></td>
</tr>
<tr>
<td>trustCertificateKeystoreType=[type]</td>
<td>Defines the KeyStore type for trusted root certificates. If type is set to NULL or empty, JKS is used by default. The standard keystore types supported by the JVM are JKS and PKCS12. Your environment may have more available depending on what security products are installed and available to the JVM.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>clientCertificateKeystoreUrl</td>
<td>Defines the URL of the client KeyStore. If none specified, Java defaults are used.</td>
</tr>
<tr>
<td></td>
<td>• Property type: <strong>string</strong></td>
</tr>
</tbody>
</table>
### Using the MySQL Enterprise Plug-in for Connector/J

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

The following example configures a Connector/J to communicate with the MySQL Enterprise Service Manager localhost, on port 18443, using the agent username `agent`, and password `PASSWORD`. Add the properties to your connection string on a single line:

```java
statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters
&serviceManagerUrl=https://localhost:18443/
&serviceManagerUser=agent
&serviceManagerPassword=PASSWORD
```

You must also add the application-specific properties to the JDBC URL. For example, the following fragment connects to the MySQL database `test` on localhost, using the user and password of `mysqltest`, while also collecting query data and sending it to the MySQL Enterprise Service Manager on localhost:

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

Connection conn = null;
...
try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
    "?user=mysqltest" +
    "&password=mysqltest" +
    "&statementInterceptors=com.mysql.etools.jdbc.StatementPerformanceCounters" +
    "&serviceManagerUrl=https://localhost:18443/" +
    "&serviceManagerUser=agent" +
    "&serviceManagerPassword=PASSWORD"
    );
    // Do something with the Connection
    ...
} catch (SQLException ex) {
    // handle any errors
}
```

**Note**

If a DataSource is in use (typically when using Glassfish, Weblogic, or Websphere), these properties must be passed as part of the URL property, they cannot be added to the DataSource configuration itself.

If an alternate logging system has not been selected for Connector/J, it is recommended that Connector/J's log factory is configured to use something other than the standard logger by adding the following property to the URL or DataSource:

- `logger=Log4JLogger` (for applications using Log4J)
- `logger=CommonsLogger` (for applications using Apache Commons Logging)
- `logger=Jdk14Logger` (for applications using Java 1.4 or later logging)
Additional configuration of the plug-in is done in the Enterprise Service Manager's user interface, using the **Query Analyzer** tab. From here, the capture of query performance data for a given MySQL instance that the plug-in is being with can be enabled or disabled. It is also possible to configure the level of detail that is captured; summary, examples (with source code locations) and **EXPLAIN** plans.

### 12.3 Using the MySQL Enterprise Plugin for Connector/NET

The MySQL Enterprise Plugin for Connector/NET enables you to use the Query Analyzer to monitor MySQL queries from any application using Connector/NET, including both standalone and web-based applications. As described in Section 29.3, “Query Analyzer User Interface”, the Query Analyzer can help you locate queries that are inefficient or slow. Tuning such queries helps to shorten load times for web pages, and improves overall system responsiveness and scalability.

Information about the queries is sent directly to the MySQL Enterprise Service Manager. Once you install the MySQL Enterprise Plugin for Connector/NET, query analysis becomes available for your applications without requiring any modification to the application code.

**Figure 12.3 Connector Plug-in Architecture**

---

**Prerequisites**

- You must be using Connector/NET version 6.2.3 or later.
- For MySQL 5.5, or earlier, the MySQL client application user must have **SELECT** privileges on the `mysql.inventory` table. This table contains the server UUID; it is required to report the Query Analyzer data to the MySQL Enterprise Service Manager. Use the **GRANT** statement. For example:

  ```sql
  mysql> GRANT SELECT on mysql.inventory to 'user'@'localhost' IDENTIFIED BY 'password';
  ```

- Your application should already be using the `Mysql.data.dll` and have been built with the library requirement.
- If you are using the released builds of Connector/NET, you must include the `logging=true` option within your connection string.
Installation

Download the MySQL Enterprise Plugin for Connector/.NET package. Extract the package using a suitable zip tool, and place the plug-in library, `MySql.MonitorPlugin.dll`, in the same directory as your compiled application.

Configuration

If the application does not have an `app.config` application configuration file, then make one.

To enable Query Analyzer functionality, register the trace listeners in the `System.Diagnostics` section of the `app.config` file. The following example shows the format of a typical configuration file:

```xml
<system.diagnostics>
  <sources>
    <source name="mysql" switchName="SourceSwitch" switchType="System.Diagnostics.SourceSwitch">
      <listeners>
        <add name="EMTrace" type="MySql.EMTrace.EMTraceListener, MySql.MonitorPlugin"
             initializeData=""
             Host="SERVERHOST:SERVERPORT"
             PostInterval="POSTINTERVAL"
             UserId="AGENTUSERID"
             Password="AGENTPASSWORD"/>
      </listeners>
    </source>
  </sources>
  <switches>
    <!-- You can set the level at which tracing is to occur -->
    <add name="SourceSwitch" value="All"/>
  </switches>
</system.diagnostics>

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

Within the configuration, set the values of the following parameters:

- **Host**
  
  The hostname and port number, separated by a colon, of the MySQL Enterprise Service Manager that receives the Query Analyzer data.

- **PostInterval**
  
  Query analyzer information is collected and then transmitted (“posted”) in a batch from your application to the MySQL Enterprise Service Manager. This value specifies the number of seconds between each transmission. Choose this value carefully. Too long and it might take some time for queries to appear in the Query Analyzer.

- **UserId**
The name of a user within MySQL Enterprise Service Manager that has rights to send agent information.

- **Password**

  The password of a user within MySQL Enterprise Service Manager that has rights to send agent information.

To get extended information on queries and have that information available through the MySQL Enterprise Monitor User Interface, enable the Connector.NET usage advisor. The extended information identifies potential issues such as a query not using an index, or not accessing all columns from a result set.

To enable the usage advisor, add `usage_advisor=true` to the connection string within your application. Enabling this option also automatically enables logging within Connector.NET. For more information, see Connector.NET 6.10 Connection-String Options Reference.

During execution of the application during development within Visual Studio, a significant amount of output is displayed in the **Output** window. To view this same trace output when running the application outside Visual Studio, configure an additional listener by adding the following within the `system.diagnostics` section of your `app.config` file:

```xml
<trace autoflush="false" indentsize="4">
  <listeners>
    <add name="consoleListener" type="System.Diagnostics.ConsoleTraceListener" />
  </listeners>
</trace>
```

**Usage**

After you set up MySQL Enterprise Plugin for Connector.NET, you monitor the performance of your .NET applications through the **Query Analyzer** tab, as described in Section 29.3, “Query Analyzer User Interface”.
Chapter 13 Uninstalling MySQL Enterprise Monitor

Table of Contents

13.1 Windows Platforms ................................................................. 113
13.2 UNIX Platforms ........................................................................ 114
13.3 Mac OS Platforms ..................................................................... 115
13.4 Unattended Uninstallations ..................................................... 116

Removing the MySQL Enterprise Monitor requires removing the MySQL Enterprise Service Manager and
the MySQL Enterprise Monitor Agent Service. In some circumstances, such as when running multiple
agents on one machine, you might remove only a single monitored server rather than the entire MySQL
Enterprise Monitor Agent Service.

13.1 Windows Platforms

Removing MySQL Enterprise Service Manager

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

If you are not saving existing data, you can delete the C:\Program Files\MySQL\Enterprise\Monitor
directory after removing MySQL Enterprise Service Manager.

Warning
If you did not remove existing data and log files when uninstalling MySQL
Enterprise Service Manager, do not remove the C:\Program Files\MySQL\Enterprise\Monitor
directory. Doing so will delete these files.

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

Removing MySQL Enterprise Monitor Agent

To remove MySQL Enterprise Monitor Agent, open the Control Panel and choose Add or Remove
Programs. Find the entry for MySQL Enterprise Monitor Agent and remove it. This executes the
uninstall program located in the C:\Program Files\MySQL\MySQL\Enterprise\Agent directory.

Warning
To remove only one of the agents from a machine that is running several agents,
do not remove the MySQL Enterprise Monitor Agent entry from the Add
or Remove Programs menu. To remove a single agent, see Removing a Single
Agent.

Removing 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 this way removes the default service. If you are running additional Monitor Agents, you must remove those agents manually. See the next section for instructions on doing this.

**Removing a Single 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 and leave other agents intact, follow these steps:

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

It is best to stop the agent before removing it; for instructions on stopping an agent see, Section 5.5.1, “Starting/Stopping the Agent on Windows”.

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

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

```
shell> sc delete AgentName
```

To confirm that the agent has been removed, check that there is no longer any entry for that agent in the Microsoft Management Console Services window.

Also remove or archive any log or configuration files associated with this agent. If you have installed any additional agents, remove them in the same way.

**13.2 UNIX Platforms**

**Removing MySQL Enterprise Service Manager**

To remove MySQL Enterprise Service Manager, find the uninstall file in the /opt/mysql/enterprise/monitor directory.

Execute this file by typing:

```
shell> ./uninstall
```

During the uninstall process you will be given the option of saving existing data and log files. Choose this option if you plan to reinstall the MySQL Enterprise Monitor.

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

**Warning**

If you did not remove existing data and log files when uninstalling the MySQL Enterprise Monitor, do not remove the /opt/mysql/enterprise/monitor directory; doing so will delete these files.
Removing MySQL Enterprise Monitor Agent

On Red Hat Enterprise Linux 4 and Fedora Core 4, the uninstall script might not stop the Tomcat server. Do this manually if necessary. To do this, see Section 4.4, “Starting/Stopping the MySQL Enterprise Monitor Services”.

Be careful not to accidentally stop any other Java processes running on your system.

On some Unix platforms, you might have to manually delete the `uninstall` application and the installation directory after you execute the uninstall process.

Removing MySQL Enterprise Monitor Agent

Prior to removal of the MySQL Enterprise Monitor Agent service, stop any agents by changing to the `init.d` directory and issuing the command `./mysql-monitor-agent stop`.

You will find the `uninstall` file in the `/opt/mysql/enterprise/agent` directory. Execute this file by navigating to this directory and typing:

```
shell> ./uninstall
```

Removing the Monitor Agent automatically deletes its associated `.log` and `.pid` files. After uninstalling the Monitor Agent, you can remove the `/opt/mysql/enterprise/agent` directory.

Removing the Monitor Agent this way removes the default service, and all the configuration files for different instances.

Removing a Single MySQL Enterprise Monitor Agent

To remove only one of the agents from a machine that is running several agents, do not run the uninstall program. To remove a single agent and leave other agents intact, follow these steps:

1. Stop the agent. See Section 5.5.3, “Starting/Stopping the Agent on Unix” for more information.
2. Confirm the location of the log files.
3. Remove the agent as a service.
4. Remove/Archive associated files.

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

To remove the agent as a daemon, remove its entry in the `init.d` directory. Also remove or archive any log or configuration files associated with this agent.

13.3 Mac OS Platforms

Removing MySQL Enterprise Service Manager

To remove the MySQL Enterprise Service Manager, run the `uninstall.app` located in the `/Applications/mysql/enterprise/monitor/` directory, or the root directory of your MySQL Enterprise Service Manager installation.

During the uninstall process you are prompted to save existing data and log files. Choose this option if you plan to reinstall the MySQL Enterprise Monitor.
Removing MySQL Enterprise Monitor Agent

Before removing MySQL Enterprise Monitor Agent, stop any running agents by changing to the init.d directory and issuing the command:

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

Run the uninstall.app file located in the /Applications/mysql/enterprise/agent directory.

Removing the Monitor Agent automatically deletes its associated .log and .pid files. After uninstalling the MySQL Enterprise Monitor Agent, you can remove the /Applications/mysql/enterprise/agent directory.

Removing the MySQL Enterprise Monitor Agent this way removes the default service, and all the configuration files for different instances.

Removing a Single MySQL Enterprise Monitor Agent

To remove an individual agent from a machine running several agents, do not run the uninstall program. To remove a single agent and leave other agents intact, 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 5.5.2, “Starting/Stopping the Agent on Mac OS X”.

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.

13.4 Unattended Uninstallations

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

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

The following options are available:
Table 13.1 MySQL Enterprise Monitor Uninstaller Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--help</td>
<td>Displays the list of options.</td>
</tr>
<tr>
<td>--version</td>
<td>Displays the product name and version.</td>
</tr>
<tr>
<td>--debuglevel &lt;debuglevel&gt;</td>
<td>Sets the verbosity of the uninstallation log. 0 is the lowest, 4 the highest, and 2 is the default.</td>
</tr>
<tr>
<td>--mode &lt;mode&gt;</td>
<td>Sets the uninstallation mode. This varies according to the platform. For example, on Linux-based systems, you can choose a GUI-based uninstaller with --mode gtk, or choose a text-only, console-based uninstallation with --mode text. The following is a list of the GUI-based uninstallation options available:</td>
</tr>
<tr>
<td></td>
<td>• Windows: Win32</td>
</tr>
<tr>
<td></td>
<td>• OS X: osx</td>
</tr>
<tr>
<td></td>
<td>• Solaris: xwindow</td>
</tr>
<tr>
<td></td>
<td>• Linux: gtk (Default) and xwindow.</td>
</tr>
<tr>
<td></td>
<td>--mode can also initiate text mode and unattended uninstallations.</td>
</tr>
<tr>
<td></td>
<td>• --mode text: starts a text-only, console-based uninstallation process. Text-based uninstallation is not available on Windows platforms.</td>
</tr>
<tr>
<td></td>
<td>• --mode unattended: starts an unattended uninstallation.</td>
</tr>
<tr>
<td>--debugtrace &lt;debugtrace&gt;</td>
<td>Sets the path and filename of the uninstallation log file.</td>
</tr>
<tr>
<td>--installer-language</td>
<td>Sets the language of the uninstallation. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• en: English. Default value.</td>
</tr>
<tr>
<td></td>
<td>• ja: Japanese.</td>
</tr>
</tbody>
</table>

Unattended Uninstallation

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

```
shell>./uninstall --mode unattended
```
Part III Using MySQL Enterprise Monitor
# Table of Contents

## 14 User Interface
14.1 Initial Log-In .......................................................... 125
14.2 Setting the Timezone and Locale ..................................... 126
14.3 Menus and Toolbars ................................................... 127
   14.3.1 Main Menus .................................................. 127
   14.3.2 Status Summary ........................................... 129
   14.3.3 System and User-defined Filters ............................ 129
14.4 What's New .......................................................... 131

## 15 Overview ............................................................. 135
15.1 Database Statistics .................................................. 135
15.2 Overview Graphs .................................................... 136
15.3 General Database Statistics ....................................... 137
15.4 Group Overview Configuration .................................... 137

## 16 MySQL Instances Dashboard ........................................... 139
16.1 MySQL Instance Dashboard UI ..................................... 139
16.2 MySQL Instance Details ........................................... 141
16.3 Adding Instances ................................................... 143
   16.3.1 Adding a MySQL Instance .................................. 143
   16.3.2 Adding Multiple MySQL Instances ......................... 147
16.4 Monitoring Amazon RDS ........................................... 147
16.5 Filtering MySQL Instances ....................................... 148

## 17 Managing Groups of Instances ....................................... 151

## 18 Enterprise Backup Dashboard ......................................... 153
18.1 Backup Overview ................................................... 153
18.2 Backup Details .................................................... 155

## 19 Replication Dashboard ................................................ 159
19.1 Replication Overview ............................................. 159
19.2 Replication Dashboard Tabs ....................................... 160
19.3 Replication Logging .............................................. 167

## 20 Reports and Graphs .................................................. 169
20.1 All Timeseries Graphs .............................................. 169
   20.1.1 Graph Controls .............................................. 169
   20.1.2 Graph Types ................................................. 171
   20.1.3 Timeseries Data Aggregation ................................. 171
20.2 Database File I/O and Lock Waits .................................. 172
   20.2.1 sys Schema .................................................. 172
   20.2.2 Database File I/O Graphs and Reports ....................... 172
   20.2.3 Lock Waits Report ........................................... 175
20.3 Processes Report ................................................... 175
20.4 Table Statistics .................................................... 176
20.5 InnoDB Buffer Pool Usage ......................................... 178

## 21 Advisors .............................................................. 179
21.1 Advisors Page ..................................................... 179
21.2 Advisor Types ..................................................... 183
21.3 Advisor Thresholds ................................................ 184
21.4 Advisor Schedules ................................................ 186

## 22 Events and Event Handlers ........................................... 187
22.1 Events ............................................................... 187
22.2 Event Handlers ..................................................... 191
   22.2.1 Event Handlers .............................................. 191
   22.2.2 Event Handlers Page ......................................... 191
### 22.3 Creating Event Handlers
- 22.3.1 Event Action Log .................................................. 198
- 22.3.2 Suspending an Event Handler ................................. 198

### 23 Expression-Based Advisor Reference ................................ 199
- 23.1 Administration Advisors .......................................... 199
- 23.2 Agent Advisors ....................................................... 205
- 23.3 Availability Advisors .............................................. 206
- 23.4 Cluster Advisors ..................................................... 207
- 23.5 Memory Usage Advisors .......................................... 209
- 23.6 Monitoring and Support Services Advisors .................... 211
- 23.7 Operating System Advisors ...................................... 212
- 23.8 Performance Advisors ............................................ 212
- 23.9 Replication Advisors ............................................. 218
- 23.10 Schema Advisors ................................................... 223
- 23.11 Security Advisors ................................................. 227

### 24 GUI-Based Advisor Reference ........................................ 235
- 24.1 Agent Health Advisor ............................................. 235
- 24.2 MySQL Enterprise Backup Health Advisor ..................... 238
- 24.3 MySQL Process Discovery Advisor ............................... 239
- 24.4 Duplicate Host Identity ........................................... 240
- 24.5 Duplicate MySQL Server UUID .................................. 240
- 24.6 sys Schema Install Advisor ........................................ 241
- 24.7 CPU Utilization Advisor ........................................... 241
- 24.8 Filesystem Free Space Advisor .................................. 242
- 24.9 MySQL Process ....................................................... 244
- 24.10 Query Analysis Advisors ......................................... 244
- 24.11 Group Replication Advisors ..................................... 246
- 24.12 Security Advisors .................................................. 248

### 25 Access Control .......................................................... 251
- 25.1 Users and Roles ...................................................... 251
- 25.2 Permissions ........................................................... 252
- 25.3 Monitored Assets Permissions ..................................... 252
  - 25.3.1 Server Group ..................................................... 253
  - 25.3.2 MySQL Instances .............................................. 253
- 25.4 Monitoring Services ................................................ 255
- 25.5 MySQL Enterprise Monitor ....................................... 255
- 25.6 Default Users and Roles ........................................... 258
- 25.7 Creating Users and Roles ........................................ 259

### 26 Access Control - Best Practices ..................................... 261
- 26.1 Open Permission Sets ............................................. 262
- 26.2 Strict Permission Set ............................................... 264

### 27 Global Settings .......................................................... 269
- 27.1 Server Locale ......................................................... 269
- 27.2 Server Hostname ..................................................... 269
- 27.3 Customize MySQL Server Name .................................. 270
- 27.4 Data Purge Behavior ............................................... 271
- 27.5 My Oracle Support Credentials .................................. 272
- 27.6 HTTP Proxy Settings .............................................. 272
- 27.7 External Authentication ........................................... 273

### 28 Customizing MySQL Enterprise Monitor ............................ 277
- 28.1 Creating Advisors and Rules ..................................... 277
  - 28.1.1 Creating Advisors ............................................ 277
  - 28.1.2 Overview of Graph Creation ............................... 278
  - 28.1.3 Overview of Advisor Creation .............................. 279
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.1.4 Variables</td>
<td>280</td>
</tr>
<tr>
<td>28.1.5 Thresholds</td>
<td>281</td>
</tr>
<tr>
<td>28.1.6 Using Strings</td>
<td>281</td>
</tr>
<tr>
<td>28.1.7 Wiki Format</td>
<td>282</td>
</tr>
<tr>
<td>28.1.8 Creating a New Advisor: An Example</td>
<td>282</td>
</tr>
<tr>
<td>28.1.9 Creating a New Graph: An Example</td>
<td>283</td>
</tr>
<tr>
<td>28.2 Custom Data Collection</td>
<td>284</td>
</tr>
<tr>
<td>28.2.1 Custom.xml</td>
<td>285</td>
</tr>
<tr>
<td>28.2.2 Queries</td>
<td>285</td>
</tr>
<tr>
<td>28.2.3 Data Collection Attributes</td>
<td>287</td>
</tr>
<tr>
<td>28.3 Event Notification Blackout Periods</td>
<td>288</td>
</tr>
<tr>
<td>28.3.1 Scripting Blackouts</td>
<td>289</td>
</tr>
</tbody>
</table>
Chapter 14 User Interface

Table of Contents

14.1 Initial Log-In ................................................................. 125
14.2 Setting the Timezone and Locale ............................................ 126
14.3 Menus and Toolbars .......................................................... 127
   14.3.1 Main Menus ............................................................ 127
   14.3.2 Status Summary ....................................................... 129
   14.3.3 System and User-defined Filters ..................................... 129
14.4 What's New ..................................................................... 131

This chapter provides an overview of the MySQL Enterprise Monitor 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 page](image)

You must perform the following tasks:

- **Create user with Manager role**: the manager user is used for the initial session, configuring the 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 25, *Access Control*. 
This creates the Manager user described in Section 3.3.3, “Users Created on First Log-in”. Record the user name and password, as these credentials are required for any future login.

- **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 25, *Access Control*.

This is the user described in Section 3.3.3, “Users Created on First Log-in”. The agent must connect to the 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 25, *Access Control*.

- **Configure Data Retention Settings**: define how long collected data must be retained. For more information on data purging, see Section 27.4, “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 10, *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 the 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.

After specifying the time zone and locale, the MySQL Enterprise Monitor User Interface displays the What's New page.

At this point the MySQL Enterprise Service Manager Repository is being monitored, and the built-in Agent is attempting to auto-discover additional MySQL instances on the host.
14.3 Menus and Toolbars

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

14.3.1 Main Menus

This section describes the main menus of the user interface.

Dashboards

- **Overview**: opens the Overview dashboard. This dashboard provides a high-level view of the current state of your monitored assets. For more information, see Chapter 15, Overview. This is the first page displayed after the initial setup steps are completed.

- **Replication**: opens the Replication dashboard. This dashboard provides a detailed view of the current state of your monitored replication servers. For more information, see Chapter 19, Replication Dashboard.

- **Enterprise Backup**: opens the Enterprise Backup dashboard. This dashboard provides a detailed view of the configured backups of your monitored instances. For more information, see Chapter 18, Enterprise Backup Dashboard.

- **MySQL Instances**: opens the MySQL Instances dashboard. This dashboard provides a detailed view of the current state of your monitored instances. It also permits the addition, removal, or editing of connections to MySQL instances. For more information, see Chapter 16, MySQL Instances Dashboard.

Events

The Events page lists all the events for the monitored assets to which you have access. See Figure 22.1, “Events Page with Filter” for more information.

Query Analyzer

Opens the Query Analyzer page. See Chapter 29, Using the Query Analyzer for more information.

Reports and Graphs

The Reports and Graphs menu contains the following links:

- **All Timeseries**: opens the All Timeseries graphs page. See Section 20.1, “All Timeseries Graphs” for more information.

- **Database File I/O**: opens the Database File I/O page. This page displays details and graphs of latency statistics taken from Performance Schema I/O event data.

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


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

This page utilizes the \textit{sys} schema and is only supported on MySQL versions 5.6 and 5.7.

- **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 \textit{sys} schema and is only supported on MySQL versions 5.6 and 5.7.

  See Section 20.4, “Table Statistics” for more information.

- **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 20.5, “InnoDB Buffer Pool Usage” for more information.

### Refresh

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

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>The pause is temporary. If the page is manually refreshed, the pause is canceled and the defined refresh behavior resumes.</td>
</tr>
</tbody>
</table>

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

### User

This section describes the contents of the User menu.

- **User Preferences**: opens the User Preferences page, enabling the user to change their username, full name, password, timezone, and locale.

- **Logout**: ends the current user’s session.

### Global Settings (Gear)

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

- **Groups**: opens the Manage Groups page.

- **Advisors**: opens the Advisors page. See Chapter 21, Advisors for more information.

- **Event Handlers**: opens the Event Handlers page. See Chapter 22, Events and Event Handlers for more information.

- **Settings**: opens the Settings page. See Chapter 27, Global Settings for more information.

- **Roles**: opens the Roles page. See Chapter 25, Access Control for more information.

- **Users**: opens the Users page. See Chapter 25, Access Control for more information.

- **Diagnostic Report**: generates and downloads the user diagnostic file. This file contains information on the application, property files, stack traces, and all log files.

  This file is intended for MySQL Support, to assist them in diagnosing any issues you may have with your installation.
14.3.2 Status Summary

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

Figure 14.2 Status Summary

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

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

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

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

- **MySQL Instances with Bad Connection Configurations**: the number of incorrectly configured instance connections.

- **MySQL Instances Unmonitored**: the number of running MySQL instances which are not currently monitored. Links to the Unmonitored MySQL Instances list on the MySQL Instances dashboard.

  See Unmonitored MySQL Instances for more information.

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

  **Note**
  The hosts and instances represented in the Status Summary depend on the permissions defined for the user. If the user is assigned to a specific group, only the issues originating from the servers in that group are displayed in the system status bar. For example, the Hosts Monitored icon only displays the total number of hosts in the group assigned to the user's role.

14.3.3 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.
System and User-defined 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.

- **Query Analyzer**: contains the following system filters:
  - **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**.

• **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.

### 14.4 What's New

The **What’s New** page provides a simplified interface for providing updates, information about what to do next, and news related to your My Oracle Support account. By default, the information is automatically updated every hour.

Access the **What’s New** from the help menu, and select **What’s New?**.

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

• **Welcome to MEM - What’s New Highlights**, including sections for **New users, Existing users**, and previous versions.

• **My Oracle Support**

• **Important Product-Related Announcements**

#### Welcome to MEM

On the left-hand side of the page, this section provides information about new features in MySQL Enterprise Monitor, what you should do next to complete your set up, and related information. Sections exist for new and upgraded installations.

For related information, see **Chapter 7, Post-installation Considerations**.

#### My Oracle Support

My Oracle Support is the channel for interacting with Oracle Support across all products. MySQL Enterprise Monitor (MEM) provides a basic integration with My Oracle Support (MOS), allowing you to see an up-to-date list of Open and waiting-on-customer support issues in the product.

In Global Settings, provide your own My Oracle Support (MOS) / Oracle SSO credentials. These are stored encrypted in the MEM repository, and are used to sign in to the MOS services. For more information about these settings, see **Chapter 27, Global Settings**.
MySQL Enterprise Monitor periodically checks the status of MOS Service Requests (SRs) associated with the account with the provided credentials.

Any currently “Open” SRs will be listed on the What's New page.

Any Open SRs that are in a “Waiting on Customer” state, those that need your attention, are also displayed on the Overview Dashboard.

In both locations, the SRs are linked to the My Oracle Support system for your convenience.

Use the Configure What's New button on the What's New page to control if the MOS check should be performed, and also to force an immediate update of the status of MOS Support Requests.

**Note**

To poll the My Oracle Support site directly, your MySQL Enterprise Service Manager must have access to the My Oracle Support website. If your MySQL Enterprise Service Manager is not directly connected to the Internet, you can set an HTTP proxy to help access the information. For more information on configuring an HTTP proxy, see Section 27.6, “HTTP Proxy Settings”.

---

**Important Product-Related Announcements**

This section provides a list of news items from a Web-based news feed. The news feed contains important information, including, but not limited to:

- New releases of MySQL Products, including MySQL Server and MySQL Enterprise Monitor.
- End-of-life and support changes.
- Urgent errors and notifications.

**General Configuration**

To configure the operation and forcibly update the information on the What's New page, click the Configure What's New button at the top-left of the What's New page.

The configurable options are:

- **Automatically check for Product-Related Announcements?**

  To disable the New Announcements updates, deselect the checkbox. Updates are enabled by default.

  The last time an update of the New Announcements information was successfully executed is displayed below the title.

  You can force an update of the New Announcements information by selecting the yes radio button next to Force check now?. A check will be executed when you click the save button to close the configuration dialog.

- **Automatically check for updates to open My Oracle Support Issues?**

  To disable the Support Issues updates, deselect the checkbox. Updates are enabled by default.

  The last time an update of the Support Issues information was successfully executed is displayed below the title.
You can force an update of the Support Issues information by selecting the yes radio button next to **Force check now?**. A check will be executed when you click the **save** button to close the configuration dialog.

Click **Save** to save your settings. If you selected force an update, the update will take place in the background. To cancel your changes, click **Cancel**.
Chapter 15 Overview

Table of Contents

15.1 Database Statistics .................................................................................................................. 135
15.2 Overview Graphs ..................................................................................................................... 136
15.3 General Database Statistics ................................................................................................... 137
15.4 Group Overview Configuration ............................................................................................... 137

The Overview Dashboard shows a high level summary of the MySQL instances and hosts that are monitored by MySQL Enterprise Monitor.

The Overview summarizes various key statistics related to the group that is selected, such Database Availability, Connections, Database Activity, Query Response Times, and any current MySQL Instances or Hosts that have active Critical or Emergency level events against them. It is designed in this way to give you a quick high level picture of assets that require immediate attention, as well as give you an up to date profile of how MySQL Instances within environment are behaving.

Figure 15.1 Overview Dashboard

15.1 Database Statistics

The Database Availability statistics show an aggregate of availability statistics, generated by the MySQL Availability Advisor, for all MySQL Instances within the selected Group, and allows you to monitor your Service Level Agreements for availability.

Note

The MySQL Availability Advisor must be enabled for this functionality to work. It is enabled by default.
Overview Graphs

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 into 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 into slices of time (1 or 2 pixels each, depending on the width of the display), which represent a period of time: Day = 4 minutes, Week = 20 minutes and Month = 2 hours. Within those slices, the time within each availability state is aggregated across all MySQL Instances within the selected Group, and shown as the percentage of the total time in the slice.

For example, if you have four MySQL Instances within the selected group, with three of them being up and one of them being down for a 4 minute period within the day, the slice representing those 4 minutes would be 75% dark green, and 25% red.

**Note**
MySQL Availability reporting is only available when using a MySQL Enterprise Monitor Agent of version 3.0.0 and above.

### 15.2 Overview Graphs

The following graphs are displayed by default:

• **Connections - All MySQL Instances**

• **Database Activity - All MySQL Instances** (Always displayed.)

• **Query Response Time Index**

The graphs are customizable. To change a graph, do the following:

1. Select the title of the graph. The graph selection drop-down list is displayed.

2. Select the required graph from the drop-down list. The graph updates to your selection and is saved for future sessions.

To move the graphs, use the move icon in the top right corner of the graph.

It is also possible to add graphs to the Overview page. To add a graph, do the following:

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

2. Select a graph. The page reloads and the selected graph is displayed.

To remove a graph, click the red X in the top-right corner of the graph. The page reloads and the selected graph is removed.
15.3 General Database Statistics

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

Current Problem MySQL Instances and Hosts Panels

The Current Problem MySQL Instances and Current Problem MySQL Hosts list the top 5 MySQL Instances and Hosts respectively, based on whether they have open events with a current status of either Emergency or Critical.

The results are sorted by the total time that each event has had those statuses, by Emergency descending, and then Critical descending. This means that the MySQL Instances, or Hosts, that have had Emergency, then Critical events open for the longest are displayed at the top of the list.

Current Emergency & Critical Events

The Current Emergency & Critical Events panel lists a stream of the currently open events with an Emergency or Critical status. These are listed separately to the Current Problem MySQL Instances or Current Problem Hosts panels, as these might not show all assets within an environment that is monitoring five or more of either asset type.

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

15.4 Group Overview Configuration

The Group Overview configuration bar enables you to select the group to view in the Overview dashboard. It also enables you to define the period for which the data is displayed.

By default, the Group Overview is set to All. To edit the overview, click Edit. The Group Overview edit bar is displayed.

Figure 15.2 Group Overview Filter Configuration

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Overview</td>
<td>Drop-down list containing all groups defined in MySQL Enterprise Service Manager</td>
</tr>
<tr>
<td>Graph Time Range</td>
<td>Drop-down lists containing the time periods to apply to the graphs. The possible values are:</td>
</tr>
<tr>
<td></td>
<td>• Interval: select the duration for the overview data. If you select 1 hour, the data collected in the last hour is displayed.</td>
</tr>
<tr>
<td></td>
<td>• From/To: select a date and time range for the overview data.</td>
</tr>
<tr>
<td></td>
<td>Graph time ranges apply to the timeseries graphs, only. They do not apply to the Availability, Problem, or Current Emergency information.</td>
</tr>
</tbody>
</table>
### Group Overview Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go</td>
<td>Applies the new configuration to the Overview dashboard.</td>
</tr>
<tr>
<td>Save as Default</td>
<td>Sets the selected group and time range as the default.</td>
</tr>
<tr>
<td></td>
<td>It is not possible to save a date range, using From/To, as the default for a group.</td>
</tr>
<tr>
<td>Reset to Default</td>
<td>Resets the group overview configuration to the previously saved values.</td>
</tr>
</tbody>
</table>

⚠️ **Important**

If the default group is deleted, an error is displayed on the Overview dashboard.
Chapter 16 MySQL Instances Dashboard

Table of Contents

16.1 MySQL Instance Dashboard UI ................................................................. 139
16.2 MySQL Instance Details ................................................................. 141
16.3 Adding Instances .............................................................................. 143
  16.3.1 Adding a MySQL Instance ......................................................... 143
  16.3.2 Adding Multiple MySQL Instances ......................................... 147
16.4 Monitoring Amazon RDS ................................................................. 147
16.5 Filtering MySQL Instances ............................................................. 148

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

The MySQL Instances Dashboard contains the following:

• **MySQL Instance Details**: this list is displayed by default. It lists all the currently monitored instances. If there is a problem with a connection to one of those instances, it is highlighted in red.
  
  For more information, see Section 16.2, “MySQL Instance Details”.

• **Unreachable Agents**: displayed only if a previously contactable agent is no longer contactable.
  
  For more information, see Unreachable Agents.

• **Bad MySQL Connections**: displayed if misconfigured connections exist.
  
  For more information, see Bad Connection Configurations.

• **Unmonitored MySQL Instances**: lists the number of MySQL instances which are available, but currently unmonitored by MySQL Enterprise Monitor.
  
  For more information, see Unmonitored MySQL Instances.

### 16.1 MySQL Instance Dashboard UI

This section describes the MySQL Instances Dashboard.

**Alert Buttons**

The alert buttons list the number of problematic instances, connections, and agents in your implementation. If a problem exists, they are displayed on the top-right side of the dashboard.

**Note**

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

• **Unmonitored MySQL Instances**: lists the number of MySQL instances which are available, but currently unmonitored by MySQL Enterprise Monitor. Click to open the Unmonitored MySQL Instances list. See Unmonitored MySQL Instances for more information.
Bad Connection Configurations

- **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 uncontacetable. 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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Host</td>
<td>Lists the hostname of the monitoring agent.</td>
</tr>
<tr>
<td>Connection Details</td>
<td>Lists the IP address defined in the connection string.</td>
</tr>
<tr>
<td>Last Error Date</td>
<td>Date and time of the last occurrence of this error.</td>
</tr>
<tr>
<td>Error Details</td>
<td>Cause of the error.</td>
</tr>
</tbody>
</table>

If the bad connection results from a misconfiguration, select **Edit Connection** from the drop-down menu. The connection configuration window is displayed, enabling you to review and edit the connection.

To delete the bad configuration, select **Delete Connection** from the drop-down menu.

### Unreachable Agents

This section lists the agents which are configured, and were communicating with the Service Manager, but cannot be contacted.

**Table 16.2 Unreachable Agents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Hostname of the server on which the agent is installed.</td>
</tr>
<tr>
<td>State</td>
<td>State of the agent. For example, if the agent is shut down properly, it signals the Service Manager that it is shutting down, and the state is displayed as <strong>SHUTDOWN</strong>. If the agent did not shutdown properly, if its host shutdown unexpectedly, or due to a network fault, the state displayed is <strong>TIMEDOUT</strong>.</td>
</tr>
<tr>
<td>Last Seen</td>
<td>Time and date at which the agent last contacted the Service Manager.</td>
</tr>
<tr>
<td>Version</td>
<td>Agent version.</td>
</tr>
<tr>
<td>UUID</td>
<td>The unique identifier of the agent.</td>
</tr>
<tr>
<td>Agent Directory</td>
<td>Agent installation directory.</td>
</tr>
</tbody>
</table>

### Unmonitored MySQL Instances

This section lists the running MySQL instances which have been detected but not added to the system. You can monitor, ignore, or cancel these connections if pending.

To begin monitoring one, or more, of the unmonitored instances, select them using the 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

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

### 16.2 MySQL Instance Details

The MySQL Instance Details section lists all the instances currently monitored by this installation and enables you to delete and edit instance configuration.

#### Editing Instances

To edit an instance, do the following:

- Select an instance by selecting the check box on the left of the instance name.
- Click Edit Instances. The Edit Instances dialog is displayed.

The Edit Instances window is identical to the Add Instance Connection window described in Section 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 drop-down menu.
Columns

The following columns are available:

Table 16.4 MySQL Instance Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>The instance names, in their assigned Groups. If no groups are defined, all MySQL instances are contained by the All group. The check box enables you to select all instances.</td>
</tr>
<tr>
<td>Notes</td>
<td>Displays a note icon, if a note was defined on the Instance Details tab. If a note was defined, hover the cursor over the note icon. The note is displayed as a tooltip.</td>
</tr>
<tr>
<td>Versions: MySQL</td>
<td>Displays the version of the monitored MySQL instance.</td>
</tr>
<tr>
<td>Versions: Agent</td>
<td>Displays the version of the monitoring agent.</td>
</tr>
<tr>
<td>Versions: Operating System</td>
<td>Displays the type and version of operating system on which the MySQL instance is installed.</td>
</tr>
<tr>
<td>Port</td>
<td>Displays the configured MySQL port.</td>
</tr>
<tr>
<td>Data Dir</td>
<td>Displays the configured data directory of the MySQL installation.</td>
</tr>
</tbody>
</table>

Group and Instance Context Menu

Note

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

The group-level context menu contains the following:

- **Support Diagnostics**: Opens the Support Diagnostics page. This enables you to generate a set of reports which you can send to MySQL Support as an attachment to a reported issue. This report can take several minutes to generate. The reports archive also includes a SQL dump of the Advisor Schedules, Inventory and Configuration schemas.

The instance-level menu contains the following:

- **Edit Instance**: opens the Edit Instance dialog.

- **Delete Instance**: deletes the instance from the MySQL Enterprise Service Manager.

  **Important**

  It is not possible to delete the MySQL Enterprise Monitor repository from the list. If you delete it, it is automatically restored to the list.

- **Refresh Inventory**: forces an inventory of the selected instance.

- **Support Diagnostics**: opens the Support Diagnostics dialog. This enables you to generate a set of reports which you can send to MySQL Support as an attachment to a reported issue. This report can take several minutes to generate. The reports archive also includes a SQL dump of the Advisor Schedules, Inventory and Configuration schemas.
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 25, 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](Image)

Table 16.5 Connection Settings Tab

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor From</td>
<td>Select an Agent from the list of Agents to monitor this MySQL database Instance.</td>
</tr>
<tr>
<td></td>
<td>It is recommend installing one Agent per Host and using that Agent to monitor all its MySQL Instances.</td>
</tr>
<tr>
<td></td>
<td>Set up the Agent as a local connection by specifying TCP/IP and an 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>If the instance is monitored remotely, it is not possible to retrieve any information on the host.</td>
</tr>
<tr>
<td></td>
<td>To monitor a host, an agent must be installed on that host.</td>
</tr>
<tr>
<td>Connect Using</td>
<td>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.</td>
</tr>
</tbody>
</table>
Adding a MySQL Instance

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Address and Port</td>
<td>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.</td>
</tr>
<tr>
<td>Admin User and Password</td>
<td>The root user, or user with the privileges defined in Creating the Admin User and the password.</td>
</tr>
<tr>
<td>Auto-Create Less Privileged Users</td>
<td>Choose Yes to create the General and Limited users on the MySQL instance. You must add a user name and password for both. For more information on these users, see Section 5.2, “Creating MySQL User Accounts for the Monitor Agent”. Choose No if you intend to use the Admin user for all data collection.</td>
</tr>
<tr>
<td>General User and Password</td>
<td>Add a user name and password for the General User.</td>
</tr>
<tr>
<td>Limited User and Password</td>
<td>Add a user name and password for the Limited User.</td>
</tr>
</tbody>
</table>

**Encryption Settings**

**Figure 16.2 Add Instance Encryption Settings**

```
Require Encryption
No

Allow Self-Signed Certificates
No

CA Certificate
```

**Note**

It is strongly recommend to use the General and Limited user.
Table 16.6 Encryption Settings Tab

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Require Encryption</td>
<td>Defines whether the connection uses TLS for security.</td>
</tr>
<tr>
<td>Allow Self-Signed Certificates</td>
<td>Specifies whether the connection permits self-signed certificates.</td>
</tr>
<tr>
<td>CA Certificate</td>
<td>Paste the CA certificate's contents here.</td>
</tr>
</tbody>
</table>

**Note**
This is not required if you are using a self-signed certificate.

Advanced Settings

Figure 16.3 Add Instance Advanced Settings

Table 16.7 Advanced Settings

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover Replication Topologies</td>
<td>Specifies whether the agent attempts to discover if the instance is part of a replication group and discover the other members of that replication group.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If set to No, no replication discovery is attempted.</td>
</tr>
<tr>
<td>MySQL Identity Source</td>
<td>Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.</td>
</tr>
</tbody>
</table>
### Adding Multiple MySQL Instances

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>uses either the server_uuid variable if present, or generates a random new UUID</td>
</tr>
<tr>
<td>Host Plus Datadir</td>
<td>generates a hash of the host identity and the path to the MySQL instances data directory to create a unique identity.</td>
</tr>
</tbody>
</table>

**Note**

Host Plus Datadir can be used only if the agent is running on the same host as the MySQL instance for this connection.

### Inventory Table Schema

When the Agent connects to the MySQL Instance, it creates an inventory table, if one does not already exist, and stores two rows within it: a generated Instance UUID, and the host ID.

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

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

- Do not configure MySQL Enterprise Monitor to auto-create the less privileged Limited and General accounts, and instead use the Admin account for all monitoring.

  This is set in the Connection Settings tab when adding or editing a MySQL instance to be monitored. The Auto-Create Less Privileged Users setting defaults to Yes, ensure it is set to No.

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

- Change the inventory table schema for MySQL Enterprise Monitor Agent from "mysql" to an existing, alternative schema.

  This is set in the Advanced Settings tab when adding (or editing) a MySQL instance to be monitored. The Inventory Table Schema setting defaults to mysql, which is typically not accessible to the Agent user in a cloud (or shared) environment. Change it to a schema you created.

Your MySQL instance is displayed on the MySQL Instances Dashboard.

### 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 MySQL Instance Details filter is displayed. For information on the named filters, see Section 14.3.3, “System and User-defined Filters”.

**Figure 16.4 MySQL Instance Filter**

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

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

### Note
Filtering on MySQL or Agent version uses a regular expression which does not support the use of partial version numbers, such as "5.". 5 or 5.6 return a result, if such versions are in use, but a partial version returns an error.
Chapter 17 Managing Groups of Instances

This chapter describes groups.

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

The primary uses for groups are:

- **Access Control**: You can assign users to specific groups. This means the user sees only those instances in the group to which they have rights. The groups are associated with Roles, and the users are assigned to the roles. For more information, see Chapter 25, *Access Control*.

- **Replication**: MySQL Enterprise Monitor automatically creates groups for replication topologies. That is, if a master-slave(s) relationship is detected, the relevant group is created to contain all members of that topology.

  **Important**
  
  It is not possible to edit replication group membership. Replication 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.

- **Convenience**: grouping related instances together, in order to ensure consistent Advisor scheduling and event generation.

### Creating Groups

**Important**

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

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

**Figure 17.1 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.
Deleting Groups

3. It is possible to create empty groups, and add the instances later, or to allow the Agent installations to add the instances to the groups by adding the group name to the **Monitor Group** field in the installer.

4. To add instances to the group, select the **Assets** tab.

5. Select the required instances by checking the check box beside the instance.

6. Click **Save** to save your new group. Click **Cancel** to discard your changes.

7. To edit a group, select it in the list, and edit as required.

![Note](icon.jpg) 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 18 Enterprise Backup Dashboard

Table of Contents

18.1 Backup Overview ................................................................. 153
18.2 Backup Details ................................................................. 155

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.

18.1 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.
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.

18.2 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**
Backup Status

Figure 18.1 Backup Details

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=xxxxxxxx
   --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 19 Replication Dashboard

Table of Contents

19.1 Replication Overview ................................................................. 159
19.2 Replication Dashboard Tabs ......................................................... 160
19.3 Replication Logging ................................................................. 167

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.

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 mysql 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 17, 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.

19.1 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 19.1 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.

19.2 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.

## Topology

The **Topology** tab graphically displays the replication topology of the selected group.

### Figure 19.2 Replication Topology Graph

This tab 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 19.3 Group Replication Single Group Topology

While the following shows normal communication between two 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.

![Group Replication Multiple Group Topology](image)

**Figure 19.4 Group Replication Multiple Group Topology**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>(All topologies) Name of the monitored instance.</td>
</tr>
<tr>
<td>Member State</td>
<td>(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.</td>
</tr>
<tr>
<td>Fetch State</td>
<td>(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.</td>
</tr>
</tbody>
</table>

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

**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.
### Status

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Apply State</strong></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Time Behind</strong></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Read Only</strong></td>
<td>(All topologies) Displays one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• <strong>ON</strong>: instance is configured as read_only</td>
</tr>
<tr>
<td></td>
<td>• <strong>SUPER</strong>: instance is configured as super_read_only</td>
</tr>
<tr>
<td></td>
<td>• <strong>OFF</strong>: instance is a source.</td>
</tr>
<tr>
<td><strong>GTID Enabled</strong></td>
<td>(Asynchronous topologies only) Displays if the server is using GTIDs.</td>
</tr>
<tr>
<td><strong>Binary Log Format</strong></td>
<td>(Asynchronous topologies only) Displays the format used by the binary log.</td>
</tr>
<tr>
<td><strong>Node Type</strong></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Flow Control</strong></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>View ID</strong></td>
<td>(Group Replication only) Unique identifier of the view.</td>
</tr>
<tr>
<td><strong>Group Auto Increment</strong></td>
<td>(Group Replication only) The auto increment defined for the group's members.</td>
</tr>
<tr>
<td><strong>Channels</strong></td>
<td>(Asynchronous topologies only) Displays the number of channels used by this instance.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>(All topologies) Displays the version of the MySQL instance.</td>
</tr>
</tbody>
</table>

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. To be listed near the top of the list, the replica must have at least 70% of all transactions performed as transactions retrieved from a source, and no more than 30% of total transactions are transactions performed directly on the replica and not originating with a source. Then higher retrieved percentage and lower errant percentage improve the ranking of the replica. Higher ranking means higher chance of promotion.

• **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.
Statistics

- **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.
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.

### 19.3 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, uncomment the following line in the *log4j.properties* file:

```
#log4j.logger.com.mysql.replication
```

and set to the following:
log4j.logger.com.mysql.replication = DEBUG, file
Chapter 20 Reports and Graphs

Table of Contents

20.1 All Timeseries Graphs .......................................................................................................................... 169
   20.1.1 Graph Controls ................................................................................................................................. 169
   20.1.2 Graph Types ................................................................................................................................. 171
   20.1.3 Timeseries Data Aggregation ........................................................................................................ 171
20.2 Database File I/O and Lock Waits ....................................................................................................... 172
   20.2.1 sys Schema ..................................................................................................................................... 172
   20.2.2 Database File I/O Graphs and Reports .................................................................................................. 172
   20.2.3 Lock Waits Report ............................................................................................................................ 175
20.3 Processes Report .................................................................................................................................... 175
20.4 Table Statistics ..................................................................................................................................... 176
20.5 InnoDB Buffer Pool Usage .................................................................................................................... 178

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

20.1 All Timeseries Graphs

This section describes the All Timeseries Graphs page.

20.1.1 Graph Controls

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

Asset Selector

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

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

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

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

Important

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

To display monitored instances only, deselect the Show All Assets check box.

**Graph Filter**

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

<table>
<thead>
<tr>
<th>Table 20.1 Timeseries Graph Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Graph Name</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>Time Range</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Filter</td>
</tr>
<tr>
<td>Save as Default</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Reset to Default</td>
</tr>
</tbody>
</table>

**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 29.3, “Query Analyzer User Interface”.

20.1.2 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 20.5, “InnoDB Buffer Pool Usage” for an example.

20.1.3 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.

![Optimizing Graph Data Status](image)

**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 the 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.

20.2 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.

20.2.1 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.

20.2.2 Database File I/O Graphs and Reports

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

Each tab contains the following common elements:

  • Show n Entries: Number of entries to show per page.
• **Search**: search the contents of the page.

• **Show/Hide Columns**: enables you to change the column set displayed on the page by selecting or deselecting the columns.

• **Page Navigation**: buttons enabling you to navigate the pages of the report.

**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 20.2 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 20.3 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 20.4 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`. 
20.2.3 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.

20.3 Processes Report

The Processes report lists the processes currently running on the selected MySQL instance, with their Thread ID, User, and so on.
20.4 Table Statistics

The Table Statistics report is based on the table statistics sys schema view. For more information, see The schema_table_statistics and x$schema_table_statistics Views.

Table View

The table view lists the statistics in a standard table format. The data can be sorted by column.
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 20.8 Table Statistics Report - Treemap View
20.5 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.INNODB_BUFFER_PAGE table, which is available in MySQL Server version 5.5.28 or higher.

Running the InnoDB Buffer Pool Usage Report

To run the usage report, do the following:

1. Navigate to the Reports & Graphs drop-down menu.
2. Select InnoDB Buffer Pool Usage.

The Generate Report page is displayed.

This page displays a warning about the table and resource requirements of the report generation process and prompts you to select a MySQL Server to run the report against.

Important

The report can take some time to return results. If no data is returned within 2 minutes, the report times out and an error is displayed.

3. Select the MySQL server from the asset tree.

The Generate Report page is displayed.

4. Click Generate Report.

The Loading buffer pool report progress message is displayed.

Note

If you click Reload while the report is generating, the report generation process is canceled and restarted. If you navigate away from the progress page, the report generation process is canceled.

5. The report is displayed.
Chapter 21 Advisors

Table of Contents

21.1 Advisors Page ....................................................................................................................... 179
21.2 Advisor Types ..................................................................................................................... 183
21.3 Advisor Thresholds ............................................................................................................. 184
21.4 Advisor Schedules .............................................................................................................. 186

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 21.1, “Advisors Page”
• Section 21.2, “Advisor Types”
• Section 21.3, “Advisor Thresholds”
• Section 21.4, “Advisor Schedules”

21.1 Advisors Page

This section describes the main Advisors page.

To display the Advisors page, select Advisors from the Configuration drop-down menu.

Figure 21.1 Advisors Page

The components and controls of the Advisors page are as follows:
### Advisor Categories

**Table 21.1 Advisor Page Controls**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Selected Advisors</td>
<td>Opens the edit dialog for the selected advisor. This control can also be used for multiple Advisors, but it is only possible to change the Schedule for multiple Advisors simultaneously. You can also edit an advisor using the drop-down menu adjacent to each advisor’s name.</td>
</tr>
<tr>
<td>Disable Selected Advisors</td>
<td>Disables all selected Advisors.</td>
</tr>
<tr>
<td>Create Advisor</td>
<td>Opens the Create Advisor page.</td>
</tr>
<tr>
<td>Import/Export</td>
<td>Opens the Custom Rule/Graph Export page. This functionality is for custom rules and graphs only.</td>
</tr>
<tr>
<td>Select All</td>
<td>Selects all Advisors.</td>
</tr>
<tr>
<td>Expand All</td>
<td>Expands all categories.</td>
</tr>
<tr>
<td>Collapse All</td>
<td>Collapses all categories and clears all selections.</td>
</tr>
<tr>
<td>Filter Advisors</td>
<td>Expands or collapses the Advisor filter. The Advisor filter enables you to filter the Advisors, groups and assets. For information on the named filters, see Section 14.3.3, “System and User-defined Filters”.</td>
</tr>
</tbody>
</table>

### Advisor Categories

The following types of Advisor are provided:

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

It is also possible to create custom Advisors.
To display the Advisors in each category, click on the Category name. For a full description of the default advisors, see Chapter 24, GUI-Based Advisor Reference and Chapter 23, 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>
<thead>
<tr>
<th>Table 21.2 Advisor Information Listing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Item</strong></td>
</tr>
<tr>
<td><strong>Info</strong></td>
</tr>
<tr>
<td><strong>Coverage</strong></td>
</tr>
<tr>
<td><strong>Schedule</strong></td>
</tr>
<tr>
<td><strong>Event Handling</strong></td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
</tr>
</tbody>
</table>

**Advisor Menu**

To open the Advisor menu, click the drop-down icon next to the Advisor’s name.

**Figure 21.2 Advisor Menu Control**

The Advisor menu is displayed:

**Figure 21.3 Advisor Pop-up Menu**

- Copy Advisor
- Edit Advisor Configuration
- Remove Advisor Configuration
- Disable Advisor
Table 21.3 Advisor Edit Menu Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Advisor</td>
<td>Opens the <strong>Create New Advisor</strong> page and appends <strong>Copy 1</strong> to the Advisor name. This enables you to define new Advisors based on existing ones.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>This option is only available for expression-based Advisors.</td>
</tr>
<tr>
<td>Edit Advisor</td>
<td>Configuration</td>
</tr>
<tr>
<td>Remove Advisor</td>
<td>Configuration</td>
</tr>
<tr>
<td>Disable Advisor</td>
<td>Disables the advisor and its associated graphs.</td>
</tr>
<tr>
<td>Delete Advisor</td>
<td>Deletes the selected advisor.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Only available for custom Advisors. It is not possible to delete the default Advisors.</td>
</tr>
</tbody>
</table>

**Group and Host Menu**

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

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

To open the Group menu, expand the Advisor and select the drop-down icon next to the Group name. The menu contains the following items:

- **Override Advisor Configuration**: opens the Advisor edit dialog, enabling you to change the Advisor’s configuration for the assets in the group. Changes made at the group level, only affect the assets within the group.

  **Important**
  
  If a host, Host1 for example, exists in multiple groups and a configuration override is applied to one of those groups, it does not affect Host1. Data is still collected and events generated for Host1 because it exists in different groups within the same advisor. To ensure the override applies to Host1, you must apply the same override to Host1 in each group which contains it.

- **Disable Advisor**: disables the Advisor for the selected group or host.
Advisor Filter

The Advisor filter enables you to search for specific Advisors, groups, hosts, or assets using text or regular expressions. To open the filter, click the filter button. For information on the named filters, see Section 14.3.3, “System and User-defined Filters”.

The Advisor Filter is displayed:

Figure 21.4 Advisor Pop-up Menu

Table 21.4 Advisor Filter Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advisors</td>
<td>Opens a drop-down menu listing all available Advisors. You can select multiple Advisors.</td>
</tr>
<tr>
<td>Groups</td>
<td>Opens a drop-down menu listing all defined groups. You can select multiple groups.</td>
</tr>
<tr>
<td>Asset Name</td>
<td>Opens a drop-down menu listing the available search types:</td>
</tr>
<tr>
<td></td>
<td>• Contains</td>
</tr>
<tr>
<td></td>
<td>• Doesn’t Contain</td>
</tr>
<tr>
<td></td>
<td>• Regex</td>
</tr>
<tr>
<td></td>
<td>• Negative Regex</td>
</tr>
<tr>
<td>Value</td>
<td>Free text field for the search term or regular expression.</td>
</tr>
<tr>
<td>Filter</td>
<td>Filters the Advisors list based on the search terms.</td>
</tr>
<tr>
<td>Filter &amp; Expand</td>
<td>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.</td>
</tr>
<tr>
<td>Discard Changes</td>
<td>Removes any changes made to the saved filter.</td>
</tr>
<tr>
<td>Reset</td>
<td>Resets all filter values.</td>
</tr>
</tbody>
</table>

21.2 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 23, *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 21.5 Agent Health - General**

GUI-based advisors are described in Chapter 24, *GUI-Based Advisor Reference*.

**21.3 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 21.6 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.
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.

21.4 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 22 Events and Event Handlers

Table of Contents

22.1 Events ........................................................................................................ 187
22.2 Event Handlers ............................................................................................ 191
   22.2.1 Event Handlers ................................................................................. 191
   22.2.2 Event Handlers Page ....................................................................... 191
22.3 Creating Event Handlers ............................................................................ 196
   22.3.1 Event Action Log .......................................................................... 198
   22.3.2 Suspending an Event Handler ....................................................... 198

This chapter describes Events and Event Handlers.

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

Event handlers define who is notified, and how they are notified, when the thresholds on Advisors are breached and how the event is treated after the status changes.

22.1 Events

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

Events are displayed on the Events page. Emergency and Critical events also appear on the Overview dashboard. The notification group or groups associated with a specific advisor receive a notification when an alert is triggered. For more information about creating notification groups, see Chapter 22, Events and Event Handlers.

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

**Important**

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

For more information, see Chapter 25, Access Control.

The following image shows the Events page with filter enabled:
Event Filter

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

Table 22.1 Events Filter Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Range</td>
<td>Enables you to choose a time range. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>All Time</strong>: filter on all events.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Interval</strong>: displays the <strong>Interval</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>From/To</strong>: displays <strong>From</strong> and <strong>To</strong> fields enabling you to define a date and time range to filter on.</td>
</tr>
<tr>
<td>State:</td>
<td>Enables you to choose the events states to filter on. The available choices are <strong>Any</strong>, <strong>Open</strong>, or <strong>Closed</strong>.</td>
</tr>
<tr>
<td>Current Status</td>
<td>Enables you to filter on specific current status.</td>
</tr>
<tr>
<td>Worst Status</td>
<td>Enables you to filter on specific worst status.</td>
</tr>
<tr>
<td>Advisors</td>
<td>Enables you to filter events based on the Advisors which generated them.</td>
</tr>
<tr>
<td>Filter</td>
<td>Click to filter the events list on the defined criteria.</td>
</tr>
<tr>
<td>Discard changes</td>
<td>Removes any changes made.</td>
</tr>
<tr>
<td>Reset to Default</td>
<td>Reverts any changes, and restores the saved criteria.</td>
</tr>
</tbody>
</table>
**Events List**

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

**Table 22.2 Events List Columns**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-select checkbox</td>
<td>Enables you to select all events.</td>
</tr>
<tr>
<td>Current</td>
<td>Displays an icon indicating the current status of the event.</td>
</tr>
<tr>
<td>Worst</td>
<td>Displays an icon indicating the worst status of the event.</td>
</tr>
<tr>
<td>Subject</td>
<td>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.</td>
</tr>
<tr>
<td>Topic</td>
<td>Displays the short description of the event.</td>
</tr>
<tr>
<td>Time</td>
<td>Displays the approximate time the event was generated.</td>
</tr>
<tr>
<td>Actions</td>
<td>Displays the possible actions. Click X to open the Close Events dialog.</td>
</tr>
</tbody>
</table>

- **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 22.3 Event State Icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Red and orange flame icon" /></td>
<td>Red and orange flame icon indicates an emergency alert.</td>
</tr>
<tr>
<td><img src="image" alt="Round red icon" /></td>
<td>Round red icon indicates a critical alert.</td>
</tr>
<tr>
<td><img src="image" alt="Triangular yellow icon" /></td>
<td>Triangular yellow icon indicates a warning.</td>
</tr>
<tr>
<td><img src="image" alt="Blue octagon with the letter &quot;i&quot;" /></td>
<td>Blue octagon with the letter &quot;i&quot; indicates an informational alert.</td>
</tr>
<tr>
<td><img src="image" alt="Green check" /></td>
<td>Green check indicates that the Advisor ran successfully and no alert was generated.</td>
</tr>
<tr>
<td><img src="image" alt="Skull icon" /></td>
<td>Skull icon indicates that the status of the Advisor is unknown.</td>
</tr>
</tbody>
</table>
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, Events and Event Handlers.

Important

Not all Advisors generate events which can be auto-closed. See Chapter 23, Expression-Based Advisor Reference and Chapter 24, 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 21.3, “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 27.4, “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.
22.2 Event Handlers

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

22.2.1 Event Handlers

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

Event handler conditions are comprised of the following elements:

- Groups of assets or individual assets.

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

- Advisors you want to raise notifications for.

- Event statuses to trigger the notifications (WARNING, CRITICAL, EMERGENCY, and so on).

The conditions are constructed in the following way:

\[
\text{Group AND Advisor AND Status}
\]

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

\[
(\text{Group A OR Group B}) \text{ AND } \text{Advisor= MySQL Process OR Advisor=CPU Utilization Advisor} \text{ AND } \text{status=Warning OR status=Critical}
\]

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.

22.2.2 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.

### 22.2.2.1 Event Handlers List

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

**Figure 22.2 Event Handlers section**

Event Handlers section contains the following controls:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create Event Handler</strong></td>
<td>Opens the <strong>Create Event Handler</strong> dialog. For more information, see Section 22.3, “Creating Event Handlers”.</td>
</tr>
<tr>
<td><strong>Show * Entries</strong></td>
<td>Select the maximum number of event handlers to display. This value is saved by default.</td>
</tr>
<tr>
<td><strong>Handler Name</strong></td>
<td>Lists the names of the event handlers.</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>Lists the state of the event handler. Possible states are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Active</strong>: the event handler is running.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Suspended</strong>: the event handler is not running.</td>
</tr>
<tr>
<td><strong>Groups</strong></td>
<td>Lists the groups assigned to the event handler.</td>
</tr>
<tr>
<td><strong>Assets</strong></td>
<td>Lists the assets assigned to the event handler.</td>
</tr>
<tr>
<td><strong>Advisors</strong></td>
<td>Lists the Advisors assigned to the event handler.</td>
</tr>
<tr>
<td><strong>Statuses</strong></td>
<td>Lists the statuses assigned to the event handler.</td>
</tr>
<tr>
<td><strong>Actions</strong></td>
<td>Lists the SMTP or SNMP actions assigned to the event handler.</td>
</tr>
<tr>
<td><strong>Search</strong></td>
<td>Enables you to search for specific event handlers.</td>
</tr>
</tbody>
</table>

**Default Auto-close Policy**

The **Default Auto-close Policy** closes events after they change status. If a threshold is defined for an advisor, and the threshold is breached, an event is displayed in the **Events** page. If it changes status to a lower priority status, or to a status without a defined threshold, the default auto-close policy closes the event.
Note

The Default Auto-close Policy event handler is the only event handler created by default.

This policy does not apply to all Advisors. Some Advisors, such as MySQL Server Has Been Restarted, are too important to auto-close.

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.

22.2.2.2 Email Notification Group Controls

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

Figure 22.3 Email Notification Groups section

Table 22.5 Email Notification Groups List Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Notification Group button</td>
<td>Opens the Create Notification Group dialog. For more information, see</td>
</tr>
<tr>
<td>Group Name</td>
<td>Lists the names of the notification groups.</td>
</tr>
<tr>
<td>Recipients</td>
<td>Lists the recipients’ email addresses.</td>
</tr>
<tr>
<td>Subject Line</td>
<td>The subject line of the notification emails.</td>
</tr>
<tr>
<td>SMS</td>
<td>Status of SMS encoding. The following values are possible:</td>
</tr>
<tr>
<td></td>
<td>• true: SMS encoding is enabled.</td>
</tr>
<tr>
<td></td>
<td>• false: SMS encoding is not enabled.</td>
</tr>
<tr>
<td>MEM Admin</td>
<td>Status of emails regarding MySQL Enterprise Monitor. The following values are possible:</td>
</tr>
<tr>
<td></td>
<td>• true: critical MySQL Enterprise Monitor emails are sent to this notification group.</td>
</tr>
<tr>
<td></td>
<td>• false: no email related to MySQL Enterprise Monitor are sent to this notification group.</td>
</tr>
</tbody>
</table>

Creating an Email Notification Group

You can define email notification groups using the Create Group dialog. To open the Create Group dialog, click Create Notification Group in the Email Notification Groups section of the Event Handlers page.
The **Email Notification Groups** contains the following controls:

**Table 22.6 Create Group Controls**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Name</td>
<td>Define the name of the notification groups.</td>
</tr>
<tr>
<td>Recipients</td>
<td>Add a comma-separated list of recipients’ email addresses.</td>
</tr>
<tr>
<td>Subject Line</td>
<td>The subject line of the notification emails. You can add some, or all, of the following variables to the subject line:</td>
</tr>
<tr>
<td></td>
<td>• {0}: replaced by the alarm level of the event. Critical, Warning, and so on.</td>
</tr>
<tr>
<td></td>
<td>• {1}: replaced by the name of the advisor which generated the alert.</td>
</tr>
<tr>
<td></td>
<td>• {2}: replaced by the name of the server which triggered the alert.</td>
</tr>
<tr>
<td>SMS</td>
<td>Use SMS encoding for this notification group.</td>
</tr>
<tr>
<td>MEM Admin</td>
<td>Identifies this group as an administration group and sends critical system messages to the defined recipients.</td>
</tr>
</tbody>
</table>

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.

### 22.2.2.3 Email Settings

The **Email Settings** section enables you to define the email configuration, such as SMTP server, username and password to use for all outgoing emails.

**Figure 22.5 Email Settings section**

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

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

**22.2.2.4 Email Notification Status**

The **Email Notification Status** section displays the success or failure of the last email sent, and an error message describing why the sending failed.

**22.2.2.5 SNMP Settings**

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

Figure 22.6 SNMP Settings section

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable SNMP Notifications</td>
<td>Activates the SNMP configuration fields.</td>
</tr>
<tr>
<td>Use SNMP v1/v2</td>
<td>Choose the version of SNMP you intend to use.</td>
</tr>
<tr>
<td>Target and Port Number</td>
<td>IP address and Port number of the system which receive the SNMP Traps.</td>
</tr>
<tr>
<td>Community String</td>
<td>SNMP community string. Default value is public.</td>
</tr>
</tbody>
</table>
| Use the remote MySQL agent host IP address as the SNMP trap agent address for Advisor traps (optional) | Defines the source IP address included in the trap.  
  - Disabled: the trap uses the IP address of the service manager.  
  - Enabled: the trap uses the IP address of the agent monitoring the host for which the advisor was triggered. |
| SNMP trap agent address for internally generated traps (optional) | Defines the source IP address included in traps generated by MySQL Enterprise Service Manager |
| On Save send test trap | Send a test trap message when Save is clicked. Select one, or more, of the trap types from the list. One trap is sent for each option selected. |

### 22.2.2.6 SNMP Notification Status

The **SNMP Notification Status** section displays the success or failure of the last trap sent, and an error message describing why the sending failed.

### 22.3 Creating Event Handlers

Event handlers enable you to create a condition which, when met, triggers notifications to concerned parties such as DBAs, System Administrators and so on.

The following condition criteria can be defined:

- Assets and Groups: enables you to select multiple assets or multiple groups to monitor.

**Important**

It is possible to define both Assets and Groups in an event handler, but is not recommended. It is recommended that you create the event handler using either
Creating Event Handlers

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.

**Table 22.9 Create Event Handler Controls**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Handler Name</td>
<td>Specify a name which uniquely identifies the new event handler.</td>
</tr>
<tr>
<td>Filters</td>
<td></td>
</tr>
<tr>
<td>Assets</td>
<td>Select the individual assets to monitor from the Assets drop-down list. If this field is left blank, all assets are included in the event handler's condition, unless one or more groups are defined. If groups are defined, and the asset field is blank, the event handler's condition includes groups only. The Assets drop-down list displays the Assets in their groups, if groups are defined. If no groups are defined, it lists the assets. It is not possible to select groups in the Assets field. You must expand the group to select individual assets.</td>
</tr>
<tr>
<td>Note</td>
<td>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.</td>
</tr>
<tr>
<td>Groups</td>
<td>Select the groups of assets to monitor. If this field is left blank, all groups are included in the event handler's condition, unless one or more assets are defined. If assets are defined, and the group field is blank, the event handler's condition includes assets only.</td>
</tr>
<tr>
<td>Advisors</td>
<td>Select the Advisors. If this field is left blank, all advisors are included in the event handler's condition.</td>
</tr>
<tr>
<td>Event Statuses</td>
<td>Select the statuses for which you want to receive notifications.</td>
</tr>
<tr>
<td>Event Handling</td>
<td></td>
</tr>
<tr>
<td>SMTP Notification Groups</td>
<td>Select the groups you want to notify.</td>
</tr>
<tr>
<td>SMTP/SMTP Notification Policy</td>
<td>Select one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>Notify on event escalation</strong>: sends a notification only if the event changes to a higher priority. From Warning to Critical, for example. No notification is sent if the event changes to a lower priority. From Critical to Warning, for example.</td>
</tr>
<tr>
<td></td>
<td><strong>Notify on any status change</strong>: sends a notification if the status changes to any other status.</td>
</tr>
</tbody>
</table>
### Event Action Log

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td><strong>Always notify</strong></td>
<td>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.</td>
</tr>
<tr>
<td>SMTP Rate Limit</td>
<td>Defines the maximum number of SMTP notifications which can be sent per minute. The default is 10.</td>
</tr>
<tr>
<td>Send SNMP Traps</td>
<td>Defines whether SNMP Traps are used for notifications.</td>
</tr>
<tr>
<td>Auto-Close Events</td>
<td>Defines whether the events are closed after the trigger status changes. The following values are possible:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Yes</strong>: the default auto-close policy is honored and the events are closed when the conditions defined are no longer met.</td>
</tr>
<tr>
<td></td>
<td>• <strong>No</strong>: the default auto-close policy is ignored and the events remain open in the Events page even after the conditions are no longer met.</td>
</tr>
</tbody>
</table>

**Important**

If you leave the Assets, Groups, and Advisors fields empty, the event handler can generate an extremely high volume of emails, depending on the number of assets monitored. It is recommended to create event handlers which address specific requirements and contain strictly defined criteria.

**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.

22.3.1 Event Action Log

If an event handler is triggered, the action taken is displayed in the Action Log [n] section at the bottom of the expanded event, where [n] represents the number of actions logged for that event.

The Action Log [n] lists the time the action was taken, the type of action (SMTP or SNMP), the failure or success of the action, and the triggering policy used to trigger the event. The Triggering Policy column lists the names of the event handlers which triggered the actions.

22.3.2 Suspending an Event Handler

To stop an event handler, select Suspend Event Handler from the event handler's drop-down menu. A prompt is displayed enabling you to enter notes on why the event handler was suspended, and confirm the suspension.

**Note**

The rights to suspend event handlers depend on the Event Blackout permission. If this right is not granted to your role, it is not possible to suspend an event handler.
This chapter describes the MySQL Enterprise Monitor expression-based Advisors.

### 23.1 Administration Advisors

This section describes the expression-based Administration Advisors.

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

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

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

Default frequency 06:00:00
Default auto-close enabled yes

Binary Log Debug Information Disabled

The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

Binary log informational events are used for debugging and related purposes. Informational events are enabled by setting the system variable binlog_rows_query_log_events=TRUE (or ON). By default, this advisor generates an event if ROW or MIXED logging is enabled and binlog_rows_query_log_events=FALSE (or OFF).

Note
Binary log informational events were introduced in MySQL 5.6.2 and are not supported by earlier versions of MySQL.

Default frequency 06:00:00
Default auto-close enabled no

Binary Logging Is Limited

The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

Binary logging can be limited to specific databases with the --binlog-do-db and the --binlog-ignore-db options. However, if these options are used, your point-in-time recovery options are limited accordingly, along with your ability to review alterations made to your system.

Default frequency 06:00:00
Default auto-close enabled yes

Binary Logging Not Enabled
The binary log captures DML, DDL, and security changes and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

**Default frequency** 06:00:00

**Default auto-close enabled** yes

### Binary Logging Not Synchronized To Disk At Each Write

By default, the binary log contents are not synchronized to disk. If the server host machine or operating system crash, there is a chance that the latest events in the binary log are not persisted on disk. You can alter this behavior using the `sync_binlog` server variable. If the value of this variable is greater than 0, the MySQL server synchronizes its binary log to disk (using `fdatasync()`) after `sync_binlog` commit groups are written to the binary log. The default value of `sync_binlog` is 0, which does no synchronizing to disk - in this case, the server relies on the operating system to flush the binary log's contents from time to time as for any other file. A value of 1 is the safest choice because in the event of a crash you lose at most one commit group from the binary log. However, it is also the slowest choice (unless the disk has a battery-backed cache, which makes synchronization very fast).

**Default frequency** 06:00:00

**Default auto-close enabled** no

### Binary Logs Automatically Removed Too Quickly

The binary log captures DML, DDL, and security changes that occur and stores these changes in a binary format. The binary log enables point-in-time recovery, preventing data loss during a disaster recovery situation. It is used on master replication servers as a record of the statements to be sent to slave servers. It also enables you to review all alterations made to your database.

However, the number of log files and the space they use can grow rapidly, especially on a busy server, so it is important to remove these files on a regular basis when they are no longer needed, as long as appropriate backups have been made. The `expire_logs_days` parameter enables automatic binary log removal.

**Default frequency** 12:00:00

**Default auto-close enabled** yes

### Database May Not Be Portable Due To Identifier Case Sensitivity

The case sensitivity of the underlying operating system determines the case sensitivity of database and table names. If you are using MySQL on only one platform, you don't normally have to worry about this. However, depending on how you have configured your server you may encounter difficulties if you want to transfer tables between platforms that differ in 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 *mysql* 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 *mysql*.

However, the general query log should not be enabled in production environments because:

- It adds overhead to the server;
- It logs statements in the order they were received, not the order they were executed, so it is not reliable for backup/recovery;
- It grows fast and can use a lot of disk space;

**Default frequency 06:00:00**

**Default auto-close enabled yes**

### Host Cache Size Not Sufficient

The MySQL server maintains a host cache in memory that contains IP address, host name, and error information about clients. It uses the host cache for several purposes:

- By caching the results of IP-to-host name lookups, the server avoids doing a DNS lookup for each client connection, thereby improving performance.
- The cache contains information about errors that occur during the connection process. Some errors are considered "blocking." If too many of these occur successively from a given host without a successful connection, the server blocks further connections from that host.

If the host cache is not large enough to handle all the hosts from which clients may connect, performance may suffer and you may lose information about client connection errors.

**Default frequency 00:05:00**

**Default auto-close enabled no**

### In-Memory Temporary Table Size Limited By Maximum Heap Table Size

If the space required to build a temporary table exceeds either `tmp_table_size` or `max_heap_table_size`, MySQL creates a disk-based table in the server's 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. The MEM Agent (in versions > 2.3.0) reads this file automatically if it exists before executing the SHOW statement.

Default frequency 00:05:00
Default auto-close enabled no

InnoDB Strict Mode Is Off

To guard against ignored typos and syntax errors in SQL, or other unintended consequences of various combinations of operational modes and SQL commands, InnoDB provides a "strict mode" of operations. In this mode, InnoDB raises error conditions in certain cases, rather than issue a warning and process the specified command (perhaps with some unintended defaults). This is analogous to 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

Next-key locking in InnoDB can be disabled, which may improve performance in some situations. However, this may result in inconsistent data when recovering from the binary logs in replication or recovery situations. You can disable most gap locks, including most next-key locks, by using `--transaction-isolation=READ-COMMITTED` or `--innodb_locks_unsafe_for_binlog=1`. Using either is perfectly safe, but only if you are also using `--binlog-format=ROW`.

Default frequency 06:00:00

Default auto-close enabled yes

No Value Set For MyISAM Recover Options

The `myisam-recover-options` option (named `myisam-recover` before MySQL 5.5.3) enables automatic MyISAM crash recovery should a MyISAM table become corrupt for some reason. If this option is not set, then a table 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
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

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

23.2 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.
23.3 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 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 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

## 23.4 Cluster Advisors

This section describes the expression-based Cluster Advisors.

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

**Cluster Data Node Data Memory Getting Low**
Cluster Data Node Has Been Restarted

Advises when the amount of Data Memory configured for the data nodes starts to run low. Database inserts start to fail as all of the memory is consumed.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Has Been Restarted

To perform useful work, the cluster data nodes must be up-and-running continuously. It is normal for a production system to run continuously for weeks, months, or longer. If a data node has been restarted recently, it may be the result of planned maintenance, but it may also be due to an unplanned event that should be investigated.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Index Memory Getting Low

Advises when the amount of Index Memory configured for the data nodes starts to run low. Database inserts start to fail as all of the memory is consumed.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Redo Buffer Space Getting Low

Advises when the redo buffers start to fill up.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Redo Log Space Getting Low

Advises when the redo log spaces start to fill up.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Undo Buffer Space Getting Low

Advises when the undo buffers start to fill up.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Data Node Undo Log Space Getting Low

Advises when the undo log spaces start to fill up.
Cluster Data Nodes Not Running

Indicates how many data nodes are not running.

Default frequency 00:05:00
Default auto-close enabled no

Cluster DiskPageBuffer Hit Ratio Is Low

Advises when the hit-rate for the DiskPageBuffer falls below a threshold. May happen temporarily after restarting one or more data nodes. This is the average ratio since the last sample period.

Default frequency 00:05:00
Default auto-close enabled no

Cluster Has Stopped

Indicates a cluster has completely stopped.

Default frequency 00:02:00
Default auto-close enabled no

23.5 Memory Usage Advisors

This section describes the expression-based Memory Usage Advisors.

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

InnoDB Buffer Cache Has Sub-Optimal Hit Rate

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

Default frequency 00:05:00
Default auto-close enabled no
Key Buffer Size May Not Be Optimal For Key Cache

The key cache hit ratio represents the proportion of keys that are being read from the key cache in memory instead of from disk. This should normally be greater than 99% for optimum efficiency.

Default frequency 00:05:00
Default auto-close enabled no

Query Cache Has Sub-Optimal Hit Rate

When enabled, the query cache should experience a high degree of "hits", meaning that queries in the cache are being reused by other user connections. A low hit rate may mean that not enough memory is allocated to the cache, identical queries are not being issued repeatedly to the server, or that the statements in the query cache are invalidated too frequently by INSERT, UPDATE or DELETE statements.

This advisor triggers when more than 25% of the Query Cache is being used, and the ratio of Query Cache hits to Query Cache inserts is low.

Default frequency 00:05:00
Default auto-close enabled no

Query Cache Potentially Undersized

When the Query Cache is full, and needs to add more queries to the cache, it makes more room in the cache by freeing the least recently used queries from the cache, and then inserting the new queries. If this is happening often then you should increase the size of the cache to avoid this constant "swapping".

Default frequency 00:05:00
Default auto-close enabled no

Table Cache Not Optimal

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

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

Default frequency 00:05:00
Default auto-close enabled no

Thread Cache Size May Not Be Optimal

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

Default frequency 00:05:00
23.6 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, select the **MEM Service Manager** item in the Asset Selector for your MySQL Enterprise Service Manager in the **All Timeseries Graphs** page.

- Checks the timestamps of data collected by the agent to ensure the time of the monitored server is not set to a future time or date. Any data collected, with a timestamp of more than 5 minutes in the future, relative to the MySQL Enterprise Service Manager's system clock, is discarded and a critical event is generated. The critical event contains information on the assets whose time is incorrectly defined.

  **Important**

  It is strongly recommended you ensure your MySQL Enterprise Service Manager server and all monitored instances synchronize their system clocks with the same time server.

- Raises a critical event if the SMTP Rate Limit defined on an Event Handler is exceeded. If this rate is exceeded, no further notifications are sent until the period ends and the new period begins (1 minute). The event lists the name of the event handler whose rate limit was exceeded and the rate defined on that event handler.

  These events are not auto-closed and are not updated. That is, they only display the first failure.

  To create an event handler which sends notifications when the SMTP Rate Limit is exceeded, in the **Create Event Handler** window, select the **ServiceManager: MEM Service Manager** asset and the **Critical** Event Status. Define other values as required.

  **Important**

  This can result in a very large volume of emails, depending on the SMTP Rate Limits defined on your Event Handlers.

**Support Diagnostics**
Unsupported Agent Version Advisor

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

23.7 Operating System Advisors

The CPU Utilization and Filesystem Free Space Advisors are described in Chapter 24, 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.

23.8 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
Binary Log Usage Exceeding Disk Cache Memory Limits

When binary log usage exceeds the binary log cache memory limits, it is performing excessive disk operations. For optimal performance, transactions that move through the binary log should be contained within the binary log cache.

**Default frequency** 00:05:00

**Default auto-close enabled** no

---

Database File I/O Global Summary

Exposes the current summary of file I/O by wait type globally via the `sys.x $io_global_by_wait_by_latency` view.

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

---

Excessive Disk Temporary Table Usage Detected

If the space required to build a temporary table exceeds either `tmp_table_size` or `max_heap_table_size`, MySQL creates a disk-based table in the server's tmpdir directory. Also, tables that have TEXT or BLOB columns are automatically placed on disk.

For performance reasons it is ideal to have most temporary tables created in memory, leaving exceedingly large temporary tables to be created on disk.

**Default frequency** 00:05:00

**Default auto-close enabled** no

---

Excessive Number of Locked Processes
Depending on the circumstances, storage engines, and other factors, one process may be using or accessing a resource (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, for example, 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, for example, 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.
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 `innodbFlushLogAtTrxCommit` to either 0 or 2. If you set the value to 2, then only an operating system crash or a power outage can erase the last second of transactions. This can be very useful on slave servers, where the loss of a second's worth of data can be recovered from the master server if needed.

Default frequency 06:00:00
Default auto-close enabled yes

InnoDB Not Using Newest File Format

InnoDB supports compressed tables (`COMPRESSED` row format) and more efficient BLOB handling (`DYNAMIC` row format), but both features require support for the latest file format (`innodb_file_format=Barracuda`). These features also require the use of the `ROW_FORMAT=[DYNAMIC|COMPRESSED]` in `CREATE TABLE` and `ALTER TABLE` statements.

Default frequency 12:00:00
Default auto-close enabled no

InnoDB Log Waits May Be Performance Bottleneck

For optimal performance, InnoDB should not have to wait before writing DML activity to the InnoDB log buffer.

Default frequency 00:05:00
MyISAM Concurrent Insert Setting May Not Be Optimal

MyISAM uses table-level locking, which can adversely affect performance when there are many concurrent INSERT and SELECT statements because INSERTs block all SELECTs until the INSERT is completed. However, MyISAM can be configured to allow INSERT and SELECT statements to run concurrently in certain situations.

- If `concurrent_insert` is set to 1 (the default, or AUTO as of MySQL 5.5.3 or later), MySQL allows INSERT and SELECT statements to run concurrently for MyISAM tables that have no free blocks in the middle of the data file.

- If `concurrent_insert` is set to 2 (available in MySQL 5.0.6 and later, or ALWAYS as of MySQL 5.5.3 or later), MySQL allows concurrent inserts for all MyISAM tables, even those that have holes. For a table with a hole, new rows are inserted at the end of the table if it is in use by another thread. Otherwise, MySQL acquires a normal write lock and inserts the row into the hole.

Setting concurrent_insert to 2 allows tables to grow even when there are holes in the middle. This can be bad for applications that delete large chunks of data but continue to issue many SELECTs, thus effectively preventing INSERTs from filling the holes.

Default frequency 06:00:00

Default auto-close enabled no

Prepared Statements Not Being Closed

Prepared statements may increase performance in applications that execute similar statements more than once, primarily because the query is parsed only once. Prepared statements can also reduce network traffic because it is only necessary to send the data for the parameters for each execution rather than the whole statement.

However, prepared statements take time to prepare and consume memory in the MySQL server until they are closed.

Default frequency 00:05:00

Default auto-close enabled no

Prepared Statements Not Being Used Effectively

Prepared statements may increase performance in applications that execute similar statements more than once, primarily because the query is parsed only once. Prepared statements can also reduce network traffic because it is only necessary to send the data for the parameters for each execution rather than the whole statement.

However, prepared statements take time to prepare and consume memory in the MySQL server until they are closed, so it is important to use them properly. If you are only executing a statement a few times, the overhead of creating a prepared statement may not be worthwhile.

Default frequency 00:05:00

Default auto-close enabled no

Query Cache Is Excessively Fragmented
Enabling the query cache can significantly increase performance for SELECT queries that are identically executed across many connections, returning the same result set. However, performance can be adversely affected if the memory used for the query cache is excessively fragmented, causing the server to pause while it is removing entries from the cache or searching the free block list for a good block to use to insert a new query into the cache.

**Default frequency** 00:05:00

**Default auto-close enabled** no

---

### Table Lock Contention Excessive

Performance can be degraded if the percentage of table operations that have to wait for a lock is high compared to the overall number of locks. This can happen when using a table-level locking storage engine, such as MyISAM, instead of a row-level locking storage engine.

**Default frequency** 00:05:00

**Default auto-close enabled** no

---

### Thread Cache Not Enabled

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

**Default frequency** 00:05:00

**Default auto-close enabled** no

---

### Thread Pool Stall Limit Too Low

The `thread_pool_stall_limit` variable enables the thread pool to handle long-running statements. If a long-running statement was permitted to block a thread group, all other connections assigned to the group are blocked and unable to start execution until the long-running statement completes.

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

23.9 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 the values `minimal` (log required columns only), `full` (log all columns), or `noblob` (log all columns except for unneeded BLOB or TEXT columns).

**Default frequency** 06:00:00  
**Default auto-close enabled** yes

### Binary Log Space Exceeds Specified Limit

The binary log is a set of files that contain information about data modifications made by the MySQL server. It enables replication as well as point-in-time recovery, preventing data loss during a disaster recovery situation. It also enables you to review all alterations made to your database.

However, binary logs can consume a very large amount of disk space and should be removed from a production server to free up space after they are no longer needed by the slaves connecting to this master server, and after they have been backed up.

**Default frequency** 06:00:00  
**Default auto-close enabled** no

### Replication Configuration Advisor
Analyzes the configuration of 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, one of the slave threads has stopped, for example, 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 way behind the position to which the I/O thread is currently writing, it is a sign that replication is getting behind and results of queries directed to the 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**: Alert 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
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 yes

Slave Relay Logs Not Automatically Purged

When a slave receives updates from its master, the I/O thread stores the data in local files known as relay logs. The slave's SQL thread reads the relay logs and executes the updates they contain. After the SQL thread has executed all the updates in a relay log, the file is no longer needed and can be deleted to conserve disk space.

Default frequency 06:00:00

Default auto-close enabled no

Slave SQL Processing Not Multi-Threaded

As of MySQL Server version 5.6, replication now supports parallel execution of transactions with multi-threading on the slave. When parallel execution is enabled, the slave SQL thread acts as the coordinator for a number of slave worker threads as determined by the value of the `slave_parallel_workers` server system variable.

The current implementation of multi-threading on the slave assumes that data and updates are partitioned on a per-database basis, and that updates within a given database occur in the same relative order as they do on the master. However, it is not necessary to coordinate transactions between different databases. Transactions can then also be distributed per database, which means that a worker thread on the slave can process successive transactions on a given database without waiting for updates to other databases to complete.

Transactions on different databases can occur in a different order on the slave than on the master, simply checking for the most recently executed transaction is not a guarantee that all previous transactions on the master have been executed on the slave. This has implications for logging and recovery when using a multi-threaded slave.

Finally, note that beginning with MySQL Server 5.7.2, there is also support for intra-schema parallelization (LOGICAL_CLOCK). See Replication Slave Options and Variables for more information.

Default frequency 06:00:00

Default auto-close enabled yes

Slave 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.
23.10 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, by inserting a NULL value into the AUTO_INCREMENT column, for example, 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.
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 06:00:00
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.

**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.

**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.
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.

Object Changed: User Has Been Dropped

23.11 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
Account Has Old Insecure Password Hash

- 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

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.

**Policy-Based Password Validation Does Not Perform Dictionary Checks**

When users create weak passwords, such as `password` or `abcd`, it compromises the security of the server, making it easier for unauthorized people to guess the password and gain access to the server. Starting with MySQL Server 5.6, MySQL offers the `validate_password` 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. Such policies as: passwords must be at least 8 characters long, have both lowercase and uppercase letters, contain at least one special non-alphanumeric character, and do not match commonly-used words.

**Default frequency** 06:00:00

**Default auto-close enabled** no

**Policy-Based Password Validation Is Weak**

When users create weak passwords, such as `password` or `abcd`, it compromises the security of the server, making it easier for unauthorized people to guess the password and gain access to the server. Starting with MySQL Server 5.6, MySQL offers the `validate_password` 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. Such policies as: 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 Plugin.

**Default frequency** 06:00:00

**Default auto-close enabled** no

**Policy-Based Password Validation Not Enabled**

When users create weak passwords, such as `password` or `abcd`, it compromises the security of the server, making it easier for unauthorized people to guess the password and gain access to the server. Starting with MySQL Server 5.6, MySQL offers the `validate_password` 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. Such policies as: 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 Plugin.

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
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 yes

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

### 24.1 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 24.1 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 24.2 Agent Health - General

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

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

  The Agent Health Advisor compares the time on the MySQL Enterprise Service Manager against the time on the monitored host. If no time-synchronization issues 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 24.3 Agent Health - Backlog

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

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

24.2 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

### 24.3 MySQL Process Discovery Advisor

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

![Important](image)

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

<table>
<thead>
<tr>
<th>Table 24.1 MySQL Process Discovery Advisor Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Attempt Connection</td>
</tr>
<tr>
<td>Alert Level</td>
</tr>
</tbody>
</table>
| Admin User | The root user of the instance, or a user that has the SUPER, CREATE and INSERT privileges on the schema in which the inventory table is created. The inventory table stores unique identifiers for the instance, and is created in the **mysql** schema by default. 

  The SUPER privilege is required to temporarily switch off replication when creating and populating the inventory table.

  If you choose to enable the **Auto-Create Less Privileged Users** option, this user is used to create those with the required privileges to monitor this instance. In this case, it also requires the PROCESS, REPLICATION CLIENT, SELECT and SHOW DATABASES privileges globally WITH GRANT OPTION. |
| Admin Password | The password for the Admin User. |
| Auto-Create Less Privileged Users | When monitoring an instance, multiple levels of user can be employed to ensure that a Process connection is not held open indefinitely. |

  • **General User**: used for general monitoring tasks that do not require SUPER level privileges, and is always connected. |

  • **Limited User**: used for potentially long running statements running with SELECT only privileges. |

  If you do not have appropriate users already, they are automatically created if this option is selected. Using these lower privileged users is recommended. |
### Name | Description
--- | ---
**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** | Choose the mechanism used to generate a unique identity for the MySQL instance if one does not already exist.

- **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

### 24.4 Duplicate Host Identity

Tracks hosts with duplicate identities (SSH keys or Windows SIDs) by analyzing changes in a host instances' networks. The advisor will trigger 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

### 24.5 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.
24.6 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.

24.7 CPU Utilization Advisor

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

**CPU Usage**

**Figure 24.4 CPU Usage**

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

**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 24.5 CPU Outliers

- **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.

### 24.8 Filesystem Free Space Advisor

Monitors and graphs the filesystem disk space usage.

**Default auto-close enabled**: yes

**General**

Figure 24.6 Filesystem - General

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

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

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

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

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

### Estimated Full Capacity

#### Figure 24.7 Filesystem - Estimated Full Capacity

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

• **Extrapolate Free Space to Zero in Graphs**: enables graphing of the projected time to full capacity, based on existing load.

• **Free Space Running Out Thresholds**: generate events based on when the free space is projected to run out.

### Percentage of Space

#### Figure 24.8 Filesystem - Percentage of Space

The **Percentage of Space** section generates events based on the percentage of free space available, relative to the total space on the monitored device.
Percentage Used in Time Range

**Figure 24.9 Filesystem - Percentage Used in Time Range**

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

### 24.9 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.

### 24.10 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_statement_history_long` be enabled in `performance_schema.setup_consumers`. This is disabled by default in MySQL 5.6.

- **Enable Example Explain**: executes EXPLAIN on the selected statement. This is executed for statements whose runtime exceeds the value defined in **Auto-Explain Threshold**.
- **Auto-Explain Threshold**: Explains are executed for statements whose runtime is longer than the value defined here.

  **Important**

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

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

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 24.10 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.

**Figure 24.11 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**.

• **Notification level when discovering a Group Replication cluster 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 cluster 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.

### 24.12 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.
Chapter 25 Access Control

Table of Contents

25.1 Users and Roles .................................................................................................................. 251
25.2 Permissions .......................................................................................................................... 252
25.3 Monitored Assets Permissions
   25.3.1 Server Group .................................................................................................................. 253
   25.3.2 MySQL Instances ......................................................................................................... 253
25.4 Monitoring Services ............................................................................................................. 255
25.5 MySQL Enterprise Monitor ................................................................................................. 255
25.6 Default Users and Roles ...................................................................................................... 258
25.7 Creating Users and Roles .................................................................................................... 259

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

25.1 Users and Roles

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

• Asset visibility: the rights to access data collected from hosts or MySQL instances. Access can be strictly limited to specific groups of monitored assets.

• Application administration: the rights to view or change the MySQL Enterprise Monitor configuration.

• Specific data access: the rights to view specific types of potentially sensitive data.

• Role reuse: rather than define permissions per user, permission sets are defined in Roles and multiple users can be assigned to each Role.

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

Roles

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

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

Users

Users are simple definitions of 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.
25.2 Permissions

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

Permission Scope

There are two distinct permissions scopes in MySQL Enterprise Monitor:

- **System-wide Permissions**: apply to all assets and groups defined on the system. System-wide roles grant access to all monitored assets.

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

---

**Important**

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

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

Permission Groupings

Permissions are grouped in the following way:

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

Permission Types

The following grant types:

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

ACL-related Error Messages

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

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

25.3 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 25.1 Core Monitored Assets

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

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

25.3.1 Server Group

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

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

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

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

25.3.2 MySQL Instances

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

- **MySQL Instances**: grants access to the data collected on the monitored MySQL Instances. Possible values are:
  - **None**: No access to MySQL Instances or the data collected on them.
  - **Read-Only**: access to the MySQL instances, but no rights to create, modify, or delete connections to those instances.
  - **Administer**: access to the MySQL instances, and can create, modify, and delete connections to those servers.
Administer is also required to access the bad connections, unreachable agents, and unmonitored instance lists on the MySQL Instances dashboard.

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

**Warning**

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

### Query Analysis Permissions

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

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

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

  - **Read-Only**: Aggregated data is presented to the user, and the Query Analyzer page is populated.

  - **Administer**: grants the right to close events containing Query Analysis aggregated data.

- **Query Analysis Example and Explain Data**: access the data for example and explain plans in the Query Analyzer. This permission depends on the Query Analysis Aggregate Data permission. This permission also defines access to events which contain EXAMPLE and EXPLAIN data. Possible values are:

  - **None**: no access is granted to the Query Analyzer EXAMPLE and EXPLAIN data.

  - **Read-Only**: EXAMPLE and EXPLAIN data is accessible. If Query Analysis Aggregate Data is not set to Read-Only, EXAMPLE and EXPLAIN data cannot be accessed.

  - **Administer**: grants the right to close events containing Query Analysis EXAMPLE and EXPLAIN data.

**Note**

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

### Monitored Asset Permission Dependencies

Each of the Monitored Asset permissions is dependent on the others. For a new role, all permissions default to None. Setting Server Group to Read-Only automatically sets all other Monitored Asset permissions to Read-Only. Similarly, if you set Server Group to Administer, MySQL Instances is also set
to Administer. It is not possible to set MySQL Instances to None if Server Group is set to Read-Only or higher.

25.4 Monitoring Services

**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.

25.5 MySQL Enterprise Monitor

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

- **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:
New Group Creation

- **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.
25.6 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**
25.7 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 25.3, “Monitored Assets Permissions” and Section 25.4, “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.
Creating a User

- **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 26 Access Control - Best Practices

Table of Contents

26.1 Open Permission Sets ........................................................................................................................................... 262
26.2 Strict Permission Set ............................................................................................................................................... 264

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 the majority of the MySQL Enterprise Monitor functionality, such as Advisors, Event Handlers, and Query Analysis.

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

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

• Group/User Administrator: responsible for user, role, and group management. This role defines who has access to MySQL Enterprise Service Manager and defines the grouping of the servers. Users in this role are typically high-level DBAs, IT administrators, or project managers. In large 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.

## 26.1 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 26.1 Manager Role Definition

<table>
<thead>
<tr>
<th>Permission</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Group</td>
<td>Administer</td>
</tr>
<tr>
<td>MySQL Instances</td>
<td>Administer</td>
</tr>
<tr>
<td>Query Analysis Aggregate Data</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Query Analysis Example and Explain Data</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Web Application Login</td>
<td>Read-Only</td>
</tr>
<tr>
<td>MySQL Enterprise Monitor</td>
<td>Administer</td>
</tr>
<tr>
<td>Advisor Configuration</td>
<td>Administer</td>
</tr>
<tr>
<td>Event Blackout</td>
<td>Administer</td>
</tr>
<tr>
<td>Event Handling</td>
<td>Administer</td>
</tr>
<tr>
<td>New Group Creation</td>
<td>Administer</td>
</tr>
<tr>
<td>Settings</td>
<td>Administer</td>
</tr>
<tr>
<td>Users and Roles</td>
<td>Administer</td>
</tr>
</tbody>
</table>

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 26.2 DBA Role Definition

<table>
<thead>
<tr>
<th>Permission</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Group</td>
<td>Administer</td>
</tr>
<tr>
<td>MySQL Instances</td>
<td>Administer</td>
</tr>
<tr>
<td>Query Analysis Aggregate Data</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Query Analysis Example and Explain Data</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Web Application Login</td>
<td>Read-Only</td>
</tr>
<tr>
<td>MySQL Enterprise Monitor</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Advisor Configuration</td>
<td>Administer</td>
</tr>
<tr>
<td>Event Blackout</td>
<td>Administer</td>
</tr>
<tr>
<td>Event Handling</td>
<td>Administer</td>
</tr>
<tr>
<td>New Group Creation</td>
<td>Administer</td>
</tr>
<tr>
<td>Settings</td>
<td>None</td>
</tr>
<tr>
<td>Users and Roles</td>
<td>None</td>
</tr>
</tbody>
</table>

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
26.2 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 26.1 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 chose 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 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 26.3 System-Wide Role Definition

<table>
<thead>
<tr>
<th>Permission</th>
<th>SeniorDBA</th>
<th>GroupAdmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Group</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>MySQL Instances</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>Query Analysis Aggregate Data</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>Query Analysis Example and Explain Data</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>Web Application Login</td>
<td>Read-Only</td>
<td>None</td>
</tr>
<tr>
<td>MySQL Enterprise Monitor</td>
<td>Administer</td>
<td>None</td>
</tr>
</tbody>
</table>
Development Group Roles

<table>
<thead>
<tr>
<th>Permission</th>
<th>SeniorDBA</th>
<th>GroupAdmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advisor Configuration</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>Event Blackout</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>Event Handling</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>New Group Creation</td>
<td>None</td>
<td>Administer</td>
</tr>
<tr>
<td>Settings</td>
<td>Administer</td>
<td>None</td>
</tr>
<tr>
<td>Users and Roles</td>
<td>None</td>
<td>Administer</td>
</tr>
</tbody>
</table>

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 26.4 Development Group Role Definition

<table>
<thead>
<tr>
<th>Permission</th>
<th>SeniorDev</th>
<th>JuniorDev</th>
<th>JuniorDBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Group</td>
<td>Administer</td>
<td>Read-Only</td>
<td>Read-Only</td>
</tr>
<tr>
<td>MySQL Instances</td>
<td>Read-Only</td>
<td>Read-Only</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Query Analysis Aggregate Data</td>
<td>Read-Only</td>
<td>Read-Only</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Query Analysis Example and Explain Data</td>
<td>Read-Only</td>
<td>Read-Only</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Web Application Login</td>
<td>Read-Only</td>
<td>Read-Only</td>
<td>Read-Only</td>
</tr>
<tr>
<td>MySQL Enterprise Monitor</td>
<td>Read-Only</td>
<td>Read-Only</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Advisor Configuration</td>
<td>Read-Only</td>
<td>Read-Only</td>
<td>Read-Only</td>
</tr>
<tr>
<td>Event Blackout</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Event Handling</td>
<td>Read-Only</td>
<td>None</td>
<td>Read-Only</td>
</tr>
<tr>
<td>New Group Creation</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Settings</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Users and Roles</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

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.
### 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 26.2 Strict Permission Set Grouped
Chapter 27 Global Settings

Table of Contents

27.1 Server Locale .................................................. 269
27.2 Server Hostname .................................................. 269
27.3 Customize MySQL Server Name .................................. 270
27.4 Data Purge Behavior ............................................. 271
27.5 My Oracle Support Credentials .................................. 272
27.6 HTTP Proxy Settings ............................................. 272
27.7 External Authentication .......................................... 273

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

27.1 Server Locale

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

27.2 Server Hostname

This section describes how to define your server hostname.

Figure 27.1 Server Hostname

Table 27.1 Server Hostname Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>Defines the hostname used in all notifications. This value must be a valid hostname.</td>
</tr>
<tr>
<td>Port Number</td>
<td>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.</td>
</tr>
<tr>
<td>Login Display Name</td>
<td>Defines the hostname displayed on the login page. It's used as an alias for the operating system hostname.</td>
</tr>
</tbody>
</table>
27.3 Customize MySQL Server Name

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

**Figure 27.2 Customize MySQL Server Name**

![Customize MySQL server name dialog](image)

**Table 27.2 Customize**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Show MySQL Server Names as:</strong></td>
<td>Defines how hostnames are displayed. The following display settings are possible:</td>
</tr>
<tr>
<td>- Reported by Agent (default)</td>
<td>the MySQL server names are displayed exactly as reported by the monitoring agent.</td>
</tr>
<tr>
<td>- Hostname only</td>
<td>only the host name is displayed, omitting suffixes such as companyname.com.</td>
</tr>
<tr>
<td>- Transformed by substitution expression</td>
<td>enables the Substitution Expression field. This enables you to replace some, or all, of the hostname with custom values.</td>
</tr>
<tr>
<td><strong>Substitution Expression</strong></td>
<td>Enables you to substitute your hostnames with custom values. These substitutions can be simple substitutions, or more complex regular expressions. The substitution expression is a name-value pair, with the original value, or regular expression on the left, and the substitution value on the right. You can include multiple, comma-separated, substitutions. For example:</td>
</tr>
<tr>
<td>dx521.example..com=Staging</td>
<td>d9x84[.]example<a href=".*">.</a>..com=Production</td>
</tr>
<tr>
<td>database-server=(.*?)[.]example..com=$1</td>
<td>&quot;database-server=&quot;&quot;, [.]example[.]com$=&quot;&quot;, dx521=&quot;Staging DB&quot;</td>
</tr>
<tr>
<td></td>
<td>More complex substitutions are possible by using a regular expression. For more information on the syntax used in these regular expressions, see <a href="#">Regular Expression Constructs</a>.</td>
</tr>
</tbody>
</table>
### Name | Description
--- | ---
Display Connection Endpoint Values | Defines how connection endpoint values are displayed. The following display settings are possible:
  - **Always (default):** endpoint values are always displayed.
  - **Never:** endpoint values are never displayed.
  - **For non-default values:** endpoint values are displayed only if they differ from 3306 and /tmp/mysql.sock.

## 27.4 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 27.3 Data Purge Behavior

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

![Data Purge Behavior](image)

**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.

### 27.5 My Oracle Support Credentials

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

**Figure 27.4 My Oracle Support Credentials**

27.6 HTTP Proxy Settings

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

**Figure 27.5 HTTP Proxy Settings**
27.7 External Authentication

Table 27.3 External Authentication

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

---

External Authentication

Enables you to configure external authentication using LDAP or Active Directory.
### Figure 27.6 External Authentication Settings: LDAP

![External Authentication Settings](image)

### Table 27.4 LDAP Authentication

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Server Hostname and Port Number</td>
<td>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.</td>
</tr>
<tr>
<td>Secondary Server Hostname and Port Number</td>
<td>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.</td>
</tr>
<tr>
<td>Connect Timeout (seconds)</td>
<td>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.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Read Timeout (seconds)</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>Encryption type required for communication with the LDAP server(s). Supported options are <strong>None</strong>, <strong>StartTLS</strong>, and <strong>SSL</strong>.</td>
</tr>
<tr>
<td><strong>Referrals</strong></td>
<td>Authentication follows the referrals provided by the server. The default is to use whatever the LDAP directory server is configured to do.</td>
</tr>
<tr>
<td><strong>External Authentication Server Allows Anonymous Binds</strong></td>
<td>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 <code>CN</code>. If the Active Directory server is not configured to honor CN binds, it cannot fetch credentials.</td>
</tr>
<tr>
<td><strong>Authentication Mode</strong></td>
<td>The authentication mode to use.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Bind as User</strong>: binds to the LDAP directory using the credentials given to login to MySQL Enterprise Service Manager</td>
</tr>
<tr>
<td></td>
<td>• <strong>Comparison</strong>: requires an LDAP login/password that can see the configured password attribute to make a comparison with the given credentials.</td>
</tr>
<tr>
<td><strong>User Full Name Attribute Name</strong></td>
<td>Define the user full name attribute. This enables the system to return the full name of the user.</td>
</tr>
<tr>
<td><strong>Search by User Distinguished Name (DN) Pattern</strong></td>
<td>In the <strong>User Search Pattern</strong> field, define the pattern specifying the LDAP search filter to use after substitution of the username, where <code>{0}</code> defines where the username should be substituted for the DN.</td>
</tr>
<tr>
<td><strong>Search by User Attribute Pattern</strong></td>
<td>In the <strong>User Search Base (leave blank for top level)</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>To search the entire subtree, starting at the User Search Base Entry, enable <strong>Search entire subtree</strong>. If disabled, a single-level search is performed, including only the top level. To include nested roles in the search, enable <strong>Search Nested Roles</strong>.</td>
</tr>
<tr>
<td><strong>User Search Attribute Pattern</strong></td>
<td>The attribute pattern to use in user searches.</td>
</tr>
<tr>
<td><strong>Map External Roles to Application Roles</strong></td>
<td>Specifies whether the roles defined in LDAP should map to MySQL Enterprise Monitor application roles. If enabled, and LDAP is not configured to be authoritative, if a user authenticates successfully via LDAP and has a valid mapped role, they are granted permissions to the application. Roles are mapped according to the entries in the Application Role/LDAP Role(s) fields, which take comma-separated lists of LDAP roles to map to the given MySQL Enterprise Monitor</td>
</tr>
</tbody>
</table>
Active Directory Authentication enables you to configure Active Directory authentication.

Table 27.5 Active Directory Authentication

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>The Active Directory Domain.</td>
</tr>
<tr>
<td>Primary Server Hostname and Port Number</td>
<td>Hostname of the Active Directory server to use.</td>
</tr>
<tr>
<td>Secondary Server Hostname and Port Number</td>
<td>Secondary Active Directory hostname. This is optional.</td>
</tr>
<tr>
<td>Map LDAP Roles to Application Roles</td>
<td>Whether the roles defined in Active Directory can be mapped to those defined in MySQL Enterprise Monitor.</td>
</tr>
</tbody>
</table>
Chapter 28 Customizing MySQL Enterprise Monitor

Table of Contents

28.1 Creating Advisors and Rules .......................................................... 277
  28.1.1 Creating Advisors .............................................................. 277
  28.1.2 Overview of Graph Creation .............................................. 278
  28.1.3 Overview of Advisor Creation ............................................ 279
  28.1.4 Variables ........................................................................... 280
  28.1.5 Thresholds .......................................................................... 281
  28.1.6 Using Strings ...................................................................... 281
  28.1.7 Wiki Format ........................................................................ 282
  28.1.8 Creating a New Advisor: An Example ................................. 282
  28.1.9 Creating a New Graph: An Example ..................................... 283
  28.2 Custom Data Collection ............................................................. 284
    28.2.1 Custom.xml ....................................................................... 285
    28.2.2 Queries ........................................................................... 285
    28.2.3 Data Collection Attributes ................................................. 287
  28.3 Event Notification Blackout Periods ........................................... 288
    28.3.1 Scripting Blackouts ............................................................ 289

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.

28.1 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.

28.1.1 Creating Advisors

Similar Advisors are grouped together in categories. To create a new Advisor, select Create Advisor on the Advisors page.

Default Advisor Categories

The following are the default Advisor categories:

- Administration
- Agent
- Availability
- Backup
- Cluster
- Graphing
- Memory Usage
- Monitoring and Support Services
Overview of Graph Creation

- 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.

28.1.2 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 28.1.9, “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.

### 28.1.3 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 21, *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. The MySQL Enterprise Monitor product uses the Java Expression Parser. The operators and functions consist of:

• The **IN()** operator.

• The MySQL functions **LEAST(), LOCATE(), ABS(), MOD(), NOW()** (returns time since Unix epoch UTC in seconds), **UNIX_TIMESTAMP** (technically a no-op), and **INTERVAL [n] SECOND, MINUTE, HOUR, WEEK, MONTH.**

Variables

- 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.

28.1.4 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:

\[
100 - (\frac{\%\text{Innodb_buffer_pool_reads}}{\%\text{Innodb_buffer_pool_read_requests}} \times 100)
\]

If the current value of `\%\text{Innodb_buffer_pool_reads}` is 4522 and the current value of `\%\text{Innodb_buffer_pool_read_requests}` is 125989, the hit ratio is 96.4%:

\[
100 - (\frac{4522}{125989} \times 100)
\]

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

Variables can be used in the Description, Advice, Action, and Links attributes of a 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.”
28.1.5 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:

\[
100 - \left( \frac{\%Threads\_created\%}{\%Connections\%} \right) \times 100 < \text{THRESHOLD}
\]

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

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

\[
\%log\_bin\% == \text{THRESHOLD}
\]

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 evaluated, to avoid false alarms. For example, the expression for the “MyISAM Key Cache Has Sub-Optimal Hit Rate” advisor is:

\[
(\%Uptime\% > 10800) \&\& (\%Key\_read\_requests\% > 10000) \\
\&\& (100 - \left( \frac{\%Key\_reads\%}{\%Key\_read\_requests\%} \right) \times 100 < \text{THRESHOLD})
\]

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.

28.1.6 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:

\[
(\%Slave\_running\% == "ON") \&\& (\%Slave\_IO\_Running\% != \text{THRESHOLD})
\]

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.
28.1.7 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>
<thead>
<tr>
<th>Markup</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>Boldface text</td>
</tr>
<tr>
<td><del>italic</del></td>
<td>Italicize text</td>
</tr>
<tr>
<td>\</td>
<td>Create a line break</td>
</tr>
<tr>
<td>\ \ \</td>
<td>Create a double line break</td>
</tr>
<tr>
<td>\ \ \ \G</td>
<td>Create a double line break</td>
</tr>
<tr>
<td>*item 1</td>
<td>Create a bulleted list item</td>
</tr>
<tr>
<td>#item 1</td>
<td>Create a numbered list item</td>
</tr>
<tr>
<td>\</td>
<td>Use the \ \ to escape special characters</td>
</tr>
<tr>
<td>'{'moreInfo:name</td>
<td>url'}'</td>
</tr>
</tbody>
</table>

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 \G;\START SLAVE; '{'moreInfo:MySQL Manual: Replication FAQ|http://dev.mysql.com/doc/refman/5.6/en/faqs-replication.html'}'`

is translated into the following HTML markup:

```html
Replication is a <b>very nice feature</b> of MySQL. Replication can be very useful for solving problems in the following areas:

* Data Distribution
* Load Balancing
* Backup and Recovery

You can check replication status and start a slave with the following commands: SHOW SLAVE STATUS \G;\START SLAVE; <a href="http://dev.mysql.com/doc/refman/5.6/en/faqs-replication.html" target="_blank">MySQL Manual: Replication FAQ</a>
```

28.1.8 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 21.3, "Advisor Edit Menu Controls".

### 28.1.9 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 28.1.2, "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 **Advisors** page, and click the **Import/Export** link. Then note the **Custom Rule/Graph/Data Items Import** section. This is where the XML file is imported.

A definition to check disk space usage may look like the following:

```xml
<?xml version="1.0"?>
<com_mysql_merlin_server_graph_Design>
  <version>1.0</version>
  <uuid>a57c2bba-ea9b-102b-b396-94aca32bee29</uuid>
  <name>Custom Graph 001</name>
</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.

28.2 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.

### 28.2.1 Custom.xml

The following XML shows the structure of a custom data collection:

```xml
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE classes SYSTEM "items-mysql-monitor.dtd">
<classes>
  <class>
    <namespace>NameSpace</namespace>
    <classname>ClassName</classname>
    <precondition><![CDATA[Add Precondition Query Here]]></precondition>
    <query><![CDATA[Add Main Query Here]]></query>
    <attributes>
      <attribute name="AttributeName1"/>
      <attribute name="AttributeName2"/>
    </attributes>
  </class>
</classes>
```

<table>
<thead>
<tr>
<th>Table 28.2 Custom Data Collection Class Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element</strong></td>
</tr>
<tr>
<td>classes</td>
</tr>
<tr>
<td>class</td>
</tr>
<tr>
<td>namespace</td>
</tr>
<tr>
<td>classname</td>
</tr>
<tr>
<td>precondition</td>
</tr>
<tr>
<td>query</td>
</tr>
<tr>
<td>attributes</td>
</tr>
</tbody>
</table>

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

### 28.2.2 Queries

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

#### 28.2.2.1 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:

```xml
<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.

### 28.2.2 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 28.2.3.2, “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:

```xml
<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.
28.2.2.3 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:

```sql
<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 28.1.7, "Wiki Format" for more information on the supported wiki markup.

28.2.3 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:

```xml
<attributes>
  <attribute name="AttributeName1" counter="true" type="INTEGER"/>
  <attribute name="AttributeName2" counter="false" type="STRING"/>
</attributes>
```

### Table 28.3 Attribute Elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the attribute defined in the AS clause of the query.</td>
</tr>
<tr>
<td>counter</td>
<td>Whether the attribute is a counter type.</td>
</tr>
<tr>
<td></td>
<td>• <strong>true</strong>: the attribute is a counter type.</td>
</tr>
<tr>
<td></td>
<td>• <strong>false</strong>: the attribute is not a counter type.</td>
</tr>
<tr>
<td>type</td>
<td>The attribute value type. Possible values are INTEGER, STRING or FLOAT.</td>
</tr>
</tbody>
</table>

**Important**

If an attribute type is incorrectly defined in the attribute definition, such as integer instead of string, it is not possible to change the value in the `custom.xml` after the agent has started. This is because it is not possible for the agent to alter attribute types after they are defined. Attempting to change it in that manner results in an `InvalidValueForTypeException` error. To correct this, you must stop the agent, edit the type definition, rename the attribute, and restart the agent.

28.2.3.1 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:
28.2.3.2 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:

```xml
<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>
```

28.3 Event Notification Blackout Periods

During maintenance periods for database servers, you can suspend Event Handlers. During such a blackout period, Event Handlers are suspended. Agents continue to collect data, data is stored in the
repository, and events are generated and displayed. Notifications, such as SNMP traps, emails and so on, are not generated.

To enable a blackout period for an individual instance, you can use the context menu on the MySQL Instances page. Open the instance menu and select Enable Event Handler Blackout. The instance name is 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
```

Check the configuration_report.txt file for the host name and port to use. Specify the correct port for the Tomcat server. Specify the server to blackout using the name that appears in the Server Tree, including the colon and port number as shown in the preceding example.

When the HTTP authentication dialog box requesting your MySQL Enterprise Monitor User Interface user name and password opens, specify the credentials for the Manager user. Use the ID and password you specified when you initially logged in to the Monitor UI.

You can also blackout a server group by entering the following URL into the address bar of your browser, substituting the appropriate host name, and server group name:

```
https://localhost:18443/rest?command=blackout&group_name=Finance&blackout_state=true
```

When the HTTP authentication dialog box opens, enter the administrator’s credentials.

To confirm that a server is blacked out, check that its name is 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.

#### 28.3.1 Scripting Blackouts

You can write a script to black out a server, rather than opening a web browser and typing entries into the address bar. This section documents a sample blackout script that can be run from the command line.

Create the following file and save it as blackout.sh or blackout.bat depending on your platform.

```
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"
```

On Unix systems, use the chmod +x blackout.sh command to make the file executable.

- **myadmin:mypassword**: the username and password of a user with blackout rights.
- **"https://servicemanager:18443/rest"**: the url and port number of the service manager.
Scripting Blackouts

- "command=blackout": establishes the context for setting the state.
- "server_name=servername:3306": Specify the server to black out using the name that appears in the Server Tree, including the colon and port number.
- "blackout_state=true": sets the blackout state of the selected server. The selected server is grayed-out in the dashboard.

To confirm that a server is blacked out, check that its name is grayed out in the Monitor UI. To end the blackout, run the same script, changing the final argument to "blackout_state=false".

Note

Restarting MySQL Enterprise Monitor does not reactivate a blacked out server.
Part IV Using the Query Analyzer
Table of Contents

29 Using the Query Analyzer .......................................................... 295
  29.1 Providing Query Analyzer Data .......................................... 296
    29.1.1 Using the MySQL Performance Schema ....................... 296
  29.2 Query Response Time index (QRTi) .................................. 299
  29.3 Query Analyzer User Interface ........................................ 300
    29.3.1 Detailed Query Information ...................................... 302
    29.3.2 Using Graphs to Identify Queries ............................... 303
    29.3.3 Filtering Query Analyzer Data .................................. 304
    29.3.4 Query Analyzer Settings ......................................... 306
    29.3.5 Exporting Query Information .................................... 307
Chapter 29 Using the Query Analyzer

Table of Contents

29.1 Providing Query Analyzer Data .......................................................... 296
  29.1.1 Using the MySQL Performance Schema ........................................... 296
29.2 Query Response Time index (QRTi) ...................................................... 299
29.3 Query Analyzer User Interface ............................................................ 300
  29.3.1 Detailed Query Information ............................................................. 302
  29.3.2 Using Graphs to Identify Queries ...................................................... 303
  29.3.3 Filtering Query Analyzer Data ......................................................... 304
  29.3.4 Query Analyzer Settings ............................................................... 306
  29.3.5 Exporting Query Information .......................................................... 307

The MySQL Query Analyzer enables you to monitor SQL statements executed on a MySQL server and see details about each query, number of executions and execution times. Similar queries with different literal values are combined for reporting purposes.

Query Analyzer collects information about SQL statements that a MySQL client application sends to the MySQL server. There are different methods that the Query Analyzer can receive this information, which are:

• Using the Performance Schema statement digests with MySQL Server 5.6.14 and above, data can be gathered directly from MySQL Server without additional configuration, using a MySQL Enterprise Monitor Agent.

• The client application can route its database requests through the Proxy and Aggregator. The Proxy routes the client's query to both the MySQL instance and the Aggregator. The Aggregator normalizes the queries and transmits them to the Service Manager.

• Install a MySQL Enterprise Monitor Plugin for a Connector that sends the information directly to MySQL Enterprise Service Manager.

Once your MySQL client application is configured to communicate via the MySQL Enterprise Monitor Agent, queries are monitored and the normalized queries are sent to the MySQL Enterprise Monitor Agent.

For the different ways to enable Query Analysis, see Section 29.1, “Providing Query Analyzer Data”. For the user interface of the Query Analyzer, see Chapter 29, Using the Query Analyzer and Section 29.3, “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 29.3, “Query Analyzer User Interface”.

Note

When MySQL Enterprise Monitor is not accessible from a Connector/J or Connector/NET query analyzer plugin, the application performance is not impacted. Over time, the plugin determines that a backlog of reportable data exists, and falls back to consolidating it over longer ranges of time. But if more than 1,000 canonical queries are being used by the application (an unlikely scenario), data is dropped.
29.1 Providing Query Analyzer Data

The MySQL Query Analyzer can be fed information from a number of different sources. The provider supplies the statistical information about the queries, execution times, result counts and other data to display and analyze on the Query Analyzer page.

There are a number of different methods available for supplying query information to MySQL Enterprise Service Manager:

• Using the Performance Schema statement digests with MySQL Server 5.6.14 and above, data can be gathered directly from MySQL Server without additional configuration.

• Using a MySQL connector with a corresponding MySQL Enterprise Monitor Plugin that provides tracing and statistical information directly to MySQL Enterprise Service Manager.

Using this method requires a connector that is capable of collecting and sending the query statistical data directly to MySQL Enterprise Service Manager. The connectors collect the basic query statistics, such as the execution time for each query, and the row counts, and provide this information to MySQL Enterprise Service Manager for analysis.

Note
This implementation type does not require the proxy component.

• Using the MySQL Enterprise Monitor Proxy and Aggregator. For more information, see Chapter 11, Proxy and Aggregator Installation.

Important
If you are using the MySQL Enterprise Monitor Proxy and Aggregator to collect query performance data, you must disable the statements_digest consumer in performance_schema.setup_consumers.

29.1.1 Using the MySQL Performance Schema

As of MySQL Enterprise Monitor 3.0.0, Query Analyzer data is automatically collected and displayed by simply monitoring MySQL Server 5.6.14 or greater, and without any additional plugins required. This ability comes from the Performance Schema Statement Digests feature (Performance Schema Statement Digests) that was added in MySQL 5.6. If you are using an earlier version of MySQL Server (5.6.13 or below), then you can continue to use a Connector Plugin or MySQL Proxy to provide performance information to the Query Analyzer.

Note
MySQL server versions prior to MySQL 5.6.14 are disabled due to a crashing bug within Statement Digests that could be triggered by collecting the data from the Agent.

Collecting Query Analyzer data from Performance Schema, rather than at the wire protocol (which is how the other sources of Query Analyzer data work) provides data about what the statements do to generate their result sets that other sources cannot provide:

• Table Lock time
• How many rows were examined versus returned
• How many temporary tables were created, and whether any were created on disk
• Whether range scans were done, and in what form they were done
• Whether sorting happened, how many rows were sorted, and what form the sort took

There is also information not available when operating in this mode that is provided to the Query Analyzer when using Connector Plugins and MySQL Proxy:
• Stack trace of where the statement originated from on the application side (Connector Plugins only)
• Histograms of response times
• Standard deviation of response times

When enabled (which is the default), the MySQL Enterprise Monitor Agent polls the `performance_schema.events_statements_summary_by_digest` table (every minute, by default) and continually compute the deltas for each of the normalized statements that are exposed during the snapshot window. This is dependent on the Performance Schema setup having the "statements_digest" consumer enabled within `performance_schema.setup_consumers`, which is enabled by default in MySQL 5.6:

```sql
mysql> SELECT * FROM performance_schema.setup_consumers WHERE name = 'statements_digest;
+-------------------+---------+
| NAME              | ENABLED |
+-------------------+---------+
| statements_digest | YES     |
+-------------------+---------+
```

If this is not enabled, then enable it with:

```sql
UPDATE performance_schema.setup_consumers SET enabled = 'YES' WHERE name = 'statements_digest';
```

**Note**
The MySQL Enterprise Monitor Agent does not TRUNCATE the `performance_schema.events_statements_summary_by_digest` table each time it reads from it, as it is possible there may be other processes/tools consuming this data. Because of this, the "Max Latency" statistic that is reported per a normalized statement within Query Analyzer is actually the maximum since either the MySQL Server started, or since a TRUNCATE TABLE `performance_schema.events_statements_summary_by_digest` was executed. This differs from the MySQL Proxy or Connector Plugins, which report the maximum run time per the aggregated snapshot period.

**Note**
The maximum space available for digest computation is 1024 bytes by default; queries exceeding this length are truncated.

As of MySQL 5.7.8, and later, and 5.6.26, and later, this value can be changed at server startup by setting the `performance_schema_max_digest_length` system variable. In MySQL
Using the MySQL Performance Schema

5.6.24, 5.6.24, 5.7.6, and 5.7.7, use `max_digest_length` instead. For MySQL 5.7 versions prior to 5.7.6, the value cannot be changed. Nor can it be changed for MySQL 5.6 versions prior to 5.6.24.

The `performance_schema.events_statements_summary_by_digest` table is a sized table in memory within the Performance Schema, and its size is auto-configured. To check the current size:

```
mysql> SHOW GLOBAL VARIABLES LIKE 'performance_schema_digests_size';
+---------------------------------+-------+
| Variable_name                   | Value |
|---------------------------------+-------|
| performance_schema_digests_size | 5000  |
+---------------------------------+-------+
```

If your application executes more than this number of normalized statements, then it is possible that you may begin losing some statement instrumentation. You can monitor this situation with the `performance_schema_digest_lost` system variable:

```
mysql> SHOW GLOBAL STATUS LIKE 'Performance_schema_digest_lost';
+--------------------------------+-------+
| Variable_name                  | Value |
|--------------------------------+-------|
| Performance_schema_digest_lost | 0     |
+--------------------------------+-------+
```

If you detect that this counter variable is growing, consider increasing the `performance_schema_digests_size` system variable. It is also possible that your statement profile has changed over time, and that you are now executing different statements than were originally tracked (this is especially possible in very long running instances). In this case, you can simply TRUNCATE TABLE `performance_schema.events_statements_summary_by_digest`, and the Query Analyzer collection automatically starts again.

When the "Example Query" feature is enabled, Query Analyzer attempts to get an example of the longest running statement during the snapshot interval by doing a LEFT JOIN with a groupwise-max on the `performance_schema.events_statements_summary_by_digest` table to the `performance_schema.events_statements_history_long` table. Using this method does not guarantee that an example statement is always provided because, by default, the `events_statements_history_long` table is a ring buffer of the last 1000 statements executed. This too differs from the Connector Plugin and MySQL Proxy sources, which always provide an example per normalized statement, per snapshot, when enabled. We collect in this way with Performance Schema to minimize load on the monitored instance rather than polling the `performance_schema.events_statements_history_long` table at too high a frequently to try and gather statistics.

**Note**

A small subset (approximately 2MB of data) of the snapshot of known prior values is retained in-memory, and the rest is spooled to disk. The spool is stored in `$MYSQL_AGENT_HOME/spool/queryAnalysis`.

The "Example Query" feature requires that the `events_statements_history_long` table is enabled within `performance_schema.setup_consumers` (this is disabled by default within MySQL 5.6):
mysql> SELECT * FROM performance_schema.setup_consumers where name = 'events_statements_history_long';
+--------------------------------+---------+
| NAME                           | ENABLED |
+--------------------------------+---------+
| events_statements_history_long | NO      |
+--------------------------------+---------+

If this is not enabled, then enable it with:

UPDATE performance_schema.setup_consumers SET enabled = 'YES' WHERE name = 'events_statements_history_long';

When "Example Query" and "Example Explain" are enabled, the MySQL Enterprise Monitor Agent attempts to run an EXPLAIN for each example statement that is discovered and ran for longer than the "Auto-Explain Threshold". Due to the way that Performance Schema exposes normalized statements, truncating any normalized statement that is longer than 1024 bytes due to memory concerns within the MySQL Server means it is possible that an EXPLAIN may fail because the truncated statements do not parse correctly when running the EXPLAIN.

29.2 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](https://en.wikipedia.org/wiki/Apdex).

How QRTi is Defined

The three measurement conditions are "optimum", "acceptable", and "unacceptable", which are defined as:

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

An example calculation

From there, we calculate an average to determine the final QRTi value. For Example, if there are 100 executions of the digested/canonical query, where 60 finished below 100ms (the optimal time frame), 30 between 100ms and 400ms (the acceptable time frame), and the remaining 10 took longer than 400ms (unacceptable time), then the QRTi score is:

\[
\frac{(60 + (\frac{30}{2}) + (10*0))}{100} = 0.75.
\]
Reading QRTi Values

The queries listed on the Query Analyzer page also have a color coded pie chart representing a breakdown of the values used in the QRTi calculation (green representing the optimal time frame, yellow the acceptable time frame, and red the unacceptable). You can mouse over the pie chart itself to see the total number of query executions that fell within each category, as well as the percentage of query executions that fell within that group.

So when doing query optimization, you want to start with the ones that have a QRTi visual pie chart that is 100% red, which means that they also have an actual QRTi value of 0. This means that “all” executions of that query took longer than the acceptable time frame (400ms by default). You can then click on the query to get more information, such as the maximum and average query times, the average number of rows examined, the average lock wait time, examine a sample query, look at an example EXPLAIN plan, see if full table scans were done, examine index usage, etc.

You can then work your way up from the queries with a QRTi value of 0, towards those that have a value of 1 (1 meaning that all instances of the query executed within the optimal time frame). Once you get to the point that you no longer have any queries with a QRTi value of less than 1, then you can go into the Query Analysis Reporting Advisor configuration, and adjust the QRTi Threshold (the target time) down, say to 50ms, and start the process all over again.

29.3 Query Analyzer User Interface

To analyze the queries captured by the agent/proxy, change to the Query Analyzer tab. The following figure shows an example of the table on that page:

Figure 29.1 MySQL Enterprise Monitor User Interface: Query Analyzer

The main Query Analyzer table provides the summary information for all of the queries executed via the agent/proxy. The table tracks all the queries submitted to the server via the agent/proxy. The table shows a maximum of 100 rows, and you can page through the list of queries by using the page numbers, or the next, previous, first, and last buttons. To filter the list of queries that are displayed, or to change the number of queries, see Section 29.3.3, “Filtering Query Analyzer Data”. To export the query information as a .csv file, see Section 29.3.5, “Exporting Query Information”. 
Each row within the table provides the statistical information for one normalized query statement. If you have configured multiple agent/proxies to accept and forward queries to different servers, then you can expand the server view. The summary information displayed is different depending on whether you have selected a server group or an individual server.

If you have selected a server group, then the information displayed is aggregated from across the entire group. The same query executed on multiple servers shows average, total and minimum/maximum information for that query across all the servers. If you select an individual server, then only queries executed on that server are included within the table.

For each row, the following columns are populated according to the selected filtering options. For example, if the filter is configured to show queries within the last 30 minutes (Interval), then only queries executed during that time are displayed, and the corresponding statistics, such as execution times, rows returned and bytes returned, reflect that 30 minute timespan.

- **Query**: The normalized version of the query. Normalization removes the query-specific data so that different queries with different data parameters are identified as the same basic query.

  The information is shown as one query per row. Each query row is expandable, and can be expanded to show the execution times for individual servers for that query.

- **Database**: The default database in use at the time of the query. The database name might be blank, or might not match the database used within the query, if you used a qualified table name (for example, `select ... from db_name.table_name`) or if you issued a `USE` statement to switch databases after connecting.

- **Execution notices**: Highlights any specific issues experienced when running queries, including excessive table scans and bad index usage. These provide an immediate indication of a problem with a query that might require additional examination.

- **Counts**: The number of times that the query has been executed. The column is sub-divided into three further columns, showing the number of executions of the query (Exec), the number of times an error has been reported for the query (Err), and the number of times a warning has been produced (Warn).

- **QRTi**: Lists the Query Response Time index of the query. For more information, see Section 29.2, “Query Response Time index (QRTi)”.

- **Latency (hh:mm:ss.ms)**: The execution time for all the matching queries. This is the time, for every invocation of the corresponding query, as calculated by comparing the time when the query was submitted and when the results were returned by the server. Times are expressed in HH:MM:SS.MS (hours, minutes, seconds, and milliseconds).

The **Exec Time** column is further subdivided into the following columns:

- **Total**: The cumulative execution time for all the executions of this query.

- **Max**: The maximum execution time for an execution of this query.

- **Avg**: The average execution time for the execution of this query.

- **Locks**: the time spent waiting for table locks caused by the query.

- **Average History graph (Avg History)**: graphs the average execution time.

When looking at the information provided in this query, compare the average and maximum execution times to see if there was a problem on a specific server or during a specific time period when the query took place, as this could indicate an issue that needs to be investigated.
Detailed Query Information

- **Rows**: The rows returned by the query. The column is sub-divided into the following columns:
  - **Total**: The sum total number of rows returned by all executions of the query.
  - **Max**: The maximum number of rows returned by a single execution of the query.
  - **Avg**: The average number of rows returned by all executions of the query.

- **Bytes**: The number of bytes returned by each query. The column is sub-divided into the following columns:
  - **Total**: The sum total bytes returned by all executions of the query.
  - **Max**: The maximum number of bytes returned by a single execution of the query.
  - **Avg**: The average number of bytes returned by all executions of the query.

- **First Seen**: The date and time the normalized version of this query was first seen, which might be earlier than the period specified by the filter.

You can sort the list of queries by clicking the column name. The direction of the sort (highest to lowest, or lowest to highest) is indicated by a triangle next to the currently selected column. The default is to sort the list of queries by the **Latency:Total** time.

To help you and locate queries you can filter the list of queries using a variety of criteria. For more information on the filtering methods available, see Section 29.3.3, “Filtering Query Analyzer Data”.

### 29.3.1 Detailed Query Information

Click on an individual query to see more detailed information about the individual query in a pop-up window. The available tabs within this window depend on whether you have configured the more detailed query information. By default, you see the **Canonical Query** view.

You can also view **Example Query**, which provides more detailed data about a specific query, including the data and parameters submitted. You can also enable **Explain Query**, which lets you remotely execute an `EXPLAIN` statement with the specified query and view the resulting information. Finally, you can view any graph data produced during the execution of the query by using the **Graphs** tab.

- **The Canonical Query** tab:

  The canonical view for a query provides three different views of the query, which can be changed using the links under the **Canonical Form** section. The **truncated** version is a shortened version of the query. The **full** version of the query is the entire query statement. Normalization removes the constants from the individual queries so that queries following the same logical structure are identified as the same basic query.

  **Note**

  The "full" version of statements provided by the digested Performance Schema may be truncated, as the Performance Schema statement digest may truncate the statement due to memory constraints.

In addition to the summary information given in the table, the **Execution Time Statistics** section provides you with more detailed execution time statistics, including the minimum time, maximum time, average time, total time and the standard deviation. The standard deviation lets you determine whether a particular invocation of a query is outside the normal distribution of times for the given query.
The **Row Statistics** provide more detailed contents on the maximum, minimum, average, total, and standard deviation for the number of rows returned by the query, and the total size and maximum size of the data returned. The time period for the total and average figures is shown under the **Time Span** header.

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

The **First Seen** reports when the normalized version of this query was first seen, whether or not this was within the indicated **Time Span**.

To close the query detail window, click the **Hide** button.

- **The Example Query** tab:

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

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

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

- **The Explain Query** tab:

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

  **Important**

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

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

  **Note**

  If the Query Analyzer is used with the MySQL Enterprise Monitor Proxy and Aggregator, **EXPLAIN** is not performed for any **SELECT** statement which uses **SQL_CALC_FOUND_ROWS**.

- **The Graphs** tab:

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

### 29.3.2 Using Graphs to Identify Queries

The MySQL Enterprise Monitor User Interface supports correlated graphs so that you can compare the graphed execution parameters, such as the server load, thread statistics, or RAM usage, against the queries that were being executed by the server or servers being monitored at that time.
You can use the correlated graphs in two different ways:

- Drag and select an area on a graph as displayed within the **Overview** dashboard, or from the **Graphs & Reports** page of the MySQL Enterprise Monitor User Interface. You can drag and select any region on a displayed graph, click the QUAN icon, and it loads the selected range into the **Query Analyzer** page, displaying the corresponding zoomed graph, and the associated queries being executed during the selected period.

- Do the same, but select the graph from within the **Query Analyzer** page.

When using the correlated graphs, selecting an area within the graph sets the start and end time within the query filtering selection. You can combine with other filtering options, such as the **Query Type**, to zero in on the queries to examine.

To use the graphs in this manner, select a starting point and click, while holding down the button, drag a selection area to set the time-range for the query display. The time range that you select is displayed above the graph as you select the area.

To export the graph, see Section 20.1, “All Timeseries Graphs”.

### 29.3.3 Filtering Query Analyzer Data

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

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

#### Basic Filter Options

The basic filter options are:

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

  **Note**

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

- **Statement Type**: Limits the search to statements of a particular type (**SELECT**, **LITERAL**, etc.).

- **DB Name**: Limits the queries to those executed within a specific database. The database match is performed using the **LIKE** match from the MySQL database, hence you can use the `%` and `_` characters to multiple and single character matches. For more information, see Pattern Matching.

- The **Time Range** menu selects whether the time selection for filtering should be based on the time **interval** (only queries recorded within the displayed time period are shown, using the **Hours** and
Filtering Query Analyzer Data

**Minutes** pop-up), or whether the selection should be based on a time period (**From/To**), where you can select the time range to be displayed.

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

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

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

**Advanced Filter Options**

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

- **Notices**: Filters on the notices column, allowing you to filter the list to show only the queries that did not raise a notice, indicated a full table scan, or indicated that a bad index was used.

- Two column filters are provided, which allow you to filter the queries based on specific values within any of the columns shown in the Query Analyzer report list.

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

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

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

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

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

**Named Filters for Query Analyzer**

To create a named 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...**.

The new named filter is marked with an asterisk (*) at the end of the name.

For more information, see **Section 14.3.3, "System and User-defined Filters"**.

Query Analyzer contains the following system filters:

- **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.
29.3.4 Query Analyzer Settings

There are a number of settings related to the Query Analyzer data. To configure the Query Analyzer, go to the Configuration, Advisors page and select the Query Analysis Advisor category. Then choose Edit Advisor Configuration from the context menu of the Query Analysis Reporting Advisor.

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

The configuration options are:

- **Enable Example Query** displays more information about individual queries. When enabled, queries and their data items (rather than the canonical form shown by default) are provided. Enabling this option could expose the full query statements and therefore could present a security issue.

  With the Example Query option enabled, an additional tab is available within the query summary details. For more information, see Section 29.3.1, “Detailed Query Information”.

  If you enable Example Query, you can also enable Example Explain. To enable this tab, select the Enable Example Explain checkbox.

- **Enable Example Explain** provides another tab when viewing a query where you can view the output from EXPLAIN output from MySQL for the selected query. This shows the full query and how the query was executed within the servers.

  Enabling this option might add overhead to the execution of your server, as the server runs an EXPLAIN statement each time it identifies a long running query.
• **Auto-Explain Threshold**: EXPLAINs are generated for queries with a runtime above this threshold. (Format: hh:mm:ss.msec)

• **QRTi Threshold**: Optimal time for response time index. For more information about QRTi, see Section 29.2, "Query Response Time index (QRTi)".

You can also define a schedule for the advisor's data collection. For more information, see Section 21.4, "Advisor Schedules"

---

**Important**

If this advisor is disabled, Query Analysis data is no longer collected from the monitored instances. The Query Analyzer continues displaying data collected prior to the advisor being disabled.

If disabled, a message is displayed on the Query Analyzer page: "N of the selected servers do not have query analyzer enabled." where N is the number of servers.

### 29.3.5 Exporting Query Information

To get the text and details of the queries displayed on the **Query Analyzer** page, click one of the icons to the right of the **Browse Queries** label in the separator bar. The query information is exported as comma-separated data in a .csv file, with fields corresponding to the columns shown in the Monitor UI, and the time of the export encoded in the filename in UTC format. The icon representing a single page exports the query information for the currently displayed page only. The icon representing a stack of pages exports the query information for all available pages of query information.

Within the **Query Analyzer** pop-up, you can also export information about each data group as a .csv text file or a .png image file, using icons next to the labels on the left side. The output file is named according to the pattern `Statement_Report_Summary_%server/group%_%creationtimestamp%.csv`. The data exported by these icons includes:

- Execution Time: Count
- Executions: Exec Time, Max Exec Time, Min Exec Time, Average Exec Time
- Rows: Rows, Max Rows, Min Rows, Average Rows
- Kilobytes: Bytes, Max Bytes, Average Bytes

---

**Note**

Microsoft Excel users on Windows users can import the .csv file as a spreadsheet. If the file contains English text, typically you can double-click it to open in Excel.

If the file contains localized Japanese text, you must use the **File > Open** menu choice within Excel to open the file.

In the Safari browser, exported files containing localized data might contain `%NN` character sequences in their names, due to browser issues with UTF-8 and Base64 encodings.
Part V Appendices
Table of Contents

A MySQL Enterprise Monitor Component Reference .......................................................... 313
  A.1 MySQL Enterprise Service Manager Reference ......................................................... 313
    A.1.1 Log Files for the MySQL Enterprise Service Manager ................................... 313
    A.1.2 The Management Information Base (MIB) File .............................................. 313
    A.1.3 The config.properties file ............................................................................... 314
  A.2 MySQL Enterprise Monitor Agent Reference ............................................................. 317
    A.2.1 Agent Log Files .............................................................................................. 317
B Managing the Inventory ................................................................................................. 319
  B.1 The Inventory Page ................................................................................................. 319
  B.2 Using the Inventory Page ....................................................................................... 320
C MySQL Enterprise Monitor Frequently Asked Questions ............................................. 321
D MySQL Enterprise Monitor Support ............................................................................ 327
  D.1 Diagnostics Report ................................................................................................. 327
Appendix A MySQL Enterprise Monitor Component Reference

Table of Contents
A.1 MySQL Enterprise Service Manager Reference ........................................................................... 313
A.1.1 Log Files for the MySQL Enterprise Service Manager ......................................................... 313
A.1.2 The Management Information Base (MIB) File .................................................................. 313
A.1.3 The config.properties file .................................................................................................. 314
A.2 MySQL Enterprise Monitor Agent Reference ............................................................................. 317
A.2.1 Agent Log Files .................................................................................................................... 317

A.1 MySQL Enterprise Service Manager Reference

A.1.1 Log Files for the MySQL Enterprise Service Manager

This section shows the location of the log files associated with the various components that make up the
MySQL Enterprise Service Manager. These files can prove useful for debugging purposes.

All log files except catalina.out are rotated to ensure they do not grow beyond 10MB in size.

Table A.1 MySQL Enterprise Service Manager Log File Locations

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

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

<table>
<thead>
<tr>
<th>Operating System</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor\support-files\MONITOR.MIB</td>
</tr>
<tr>
<td>UNIX/Linux</td>
<td>/opt/mysql/enterprise/monitor/support-files/MONITOR.MIB</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/monitor/support-files/MONITOR.MIB</td>
</tr>
</tbody>
</table>
### 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>
<thead>
<tr>
<th>Operating System</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>C:\Program Files\MySQL\Enterprise\Monitor\apache-tomcat\webapps\ROOT\WEB-INF</td>
</tr>
<tr>
<td>UNIX/Linux</td>
<td>/opt/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>/Applications/mysql/enterprise/monitor/apache-tomcat/webapps/ROOT/WEB-INF</td>
</tr>
</tbody>
</table>

Make sure that the file is secured at the filesystem level so that it cannot be read by anybody but the administrator, or MySQL Enterprise Monitor.

#### Usage

The following examples shows a generated `config.properties` file:

```properties
#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=BMcsacZdrMmM7mrnFExUFExURHdxp4C3hcrZyxcppC2QhiE=
mysql.verify_server_cert=false
mysql.server=localhost
mysql.db=mem
```

**Note**

The `mysql.pass` is encrypted.

The application has two connection pools, one to service agent traffic, and the other for the UI. You can configure them as one logical pool with a 85/15 (agent/ui) percentage split, and use "dbPool" as the pool name in the further settings. Or, you can configure each pool separately, where the pool names are "default" and "ui". Note that the names after the "." come verbatim from DBCP at [http://commons.apache.org/proper/commons-dbcp/configuration.html](http://commons.apache.org/proper/commons-dbcp/configuration.html).

#### Table A.4 Optional `config.properties` Values

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpool.default.initialSize</td>
<td>Integer</td>
<td>20</td>
</tr>
<tr>
<td>dbpool.default.maxActive</td>
<td>Integer</td>
<td>70</td>
</tr>
<tr>
<td>dbpool.default.maxIdle</td>
<td>Integer</td>
<td>20</td>
</tr>
<tr>
<td>dbpool.default.maxWaitMillis</td>
<td>String</td>
<td>30 seconds</td>
</tr>
</tbody>
</table>
The *config.properties* file

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dbpool.default.minEvictableIdleTimeMillis</code></td>
<td>String</td>
<td>15 seconds</td>
</tr>
<tr>
<td><code>dbpool.default.minIdle</code></td>
<td>Integer</td>
<td>0</td>
</tr>
<tr>
<td><code>dbpool.default.timeBetweenEvictionRunsMillis</code></td>
<td>String</td>
<td>5 seconds</td>
</tr>
<tr>
<td><code>dbpool.ui.initialSize</code></td>
<td>Integer</td>
<td>5</td>
</tr>
<tr>
<td><code>dbpool.ui.maxActive</code></td>
<td>Integer</td>
<td>15</td>
</tr>
<tr>
<td><code>dbpool.ui.maxIdle</code></td>
<td>Integer</td>
<td>5</td>
</tr>
<tr>
<td><code>dbpool.ui.maxWaitMillis</code></td>
<td>String</td>
<td>30000 seconds</td>
</tr>
<tr>
<td><code>dbpool.ui.minEvictableIdleTimeMillis</code></td>
<td>String</td>
<td>15 seconds</td>
</tr>
<tr>
<td><code>dbpool.ui.minIdle</code></td>
<td>Integer</td>
<td>0</td>
</tr>
<tr>
<td><code>dbpool.ui.timeBetweenEvictionRunsMillis</code></td>
<td>String</td>
<td>5000 milliseconds</td>
</tr>
</tbody>
</table>

- **dbpool.ui.initialSize(5)**
  
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>5</td>
</tr>
</tbody>
</table>

  The initial number of connections that are created when the pool is started.

- **dbpool.ui.maxActive(15)**
  
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>15</td>
</tr>
</tbody>
</table>

  The maximum number of active connections that can be allocated from this pool at the same time.

- **dbpool.ui.minIdle(0)**
  
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>0</td>
</tr>
</tbody>
</table>

  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)**
  
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>5</td>
</tr>
</tbody>
</table>

  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)**
The **config.properties** file

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>30000 seconds</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>5000 milliseconds</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>15 seconds</td>
</tr>
</tbody>
</table>

The minimum amount of time an object can remain idle in the pool before it is eligible for eviction.

- **dbpool.default.initialSize**: (20)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>20</td>
</tr>
</tbody>
</table>

The initial number of connections created when the pool is started.

- **dbpool.default.maxActive**: (70)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>70</td>
</tr>
</tbody>
</table>

The maximum number of active connections allocated from this pool at the same time.

- **dbpool.default.minIdle**: (0)

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>0</td>
</tr>
</tbody>
</table>

The minimum number of established connections kept in the pool at all times.

- **dbpool.default.maxIdle**: (5)
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Default Value</td>
<td>20</td>
</tr>
</tbody>
</table>

The maximum number of connections kept in the pool at all times.

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>30 seconds</td>
</tr>
</tbody>
</table>

The maximum number of milliseconds the pool waits for a connection to be returned before throwing an exception.

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>5 seconds</td>
</tr>
</tbody>
</table>

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)`

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>String</td>
</tr>
<tr>
<td>Default Value</td>
<td>15 seconds</td>
</tr>
</tbody>
</table>

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

The MySQL Enterprise Monitor Agent provides all of the information from a monitored MySQL server to MySQL Enterprise Service Manager.

The MySQL Enterprise Monitor Agent is configured through the MySQL Enterprise Monitor User Interface, and the bundled `agent.sh/agent.bat` script. Using these methods is recommended,

### Note

In MEM versions before 3.0.0, the Agent was configured using the `mysql-monitor-agent.ini` and `agent-instance.ini` configuration files.

### A.2.1 Agent Log Files

The Agent has two log files. `mysql-monitor-agent.log` is the general log, and `mysql-monitor-agent-full.log` is the full log that also contains stack traces that are useful to the Support team.
The default path to the Agent log files are as follows:

- **Windows Path:** `C:\Program Files\MySQL\Enterprise\Agent\logs\`
- **Linux Path:** `/opt/mysql/enterprise/agent/logs/`
- **Mac OS X Path:** `~/Applications/mysql/enterprise/agent/logs/`

The log files are managed with **log4j**, which is configured using **log4j.properties**. The Agent watches for changes every 60 seconds, and updates MySQL Enterprise Monitor accordingly. The default file location:

- **Windows Path:** `C:\Program Files\MySQL\Enterprise\Agent\etc\log4j.properties`
- **Linux Path:** `/opt/mysql/enterprise/agent/etc/log4j.properties`
- **Mac OS X Path:** `~/Applications/mysql/enterprise/agent/etc/log4j.properties`

The maximum size of a log file may be limited to 2GB. If MySQL Enterprise Monitor Agent cannot add information to the configured logfile, information is sent to the standard output instead.

Because the log files can become large, you could rotate the logs by defining log4j options. For example, to implement a rotation of 10 x 10MB log files:

```
log4j.appender.file.MaximumFileSize = 10MB
log4j.appender.file.MaxBackupIndex = 10
log4j.appender.file.Append = true
```

For additional information about log4j, read the log4j documentation at [http://logging.apache.org/log4j/](http://logging.apache.org/log4j/).
Appendix B Managing the Inventory

Table of Contents

B.1 The Inventory Page ................................................................. 319
B.2 Using the Inventory Page .......................................................... 320

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 http://www.mysql.com/products/.

FAQ Categories

• Security
• General Usage
• MySQL Monitor
• MySQL Query Analyzer

Security

Questions

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

Questions and Answers

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

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

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

General Usage

Questions

• C.1: How do I find Ignored MySQL Instances? And how to I show them again?

• C.2: 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.etools.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.etools.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.etools.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.etools.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. Xxxxxxx 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](http://www.mysql.com/products/enterprise/monitor-features.html).

C.2: What are the immediate benefits of implementing the MySQL Enterprise Monitor?

Often MySQL installations are implemented with default settings that may not be best suited for specific applications or usage patterns. The MySQL Advisors go to work immediately in these environments to identify potential problems and proactively notify and advise DBAs on key MySQL settings that can be tuned to improve availability, tighten security, and increase the throughput of their existing MySQL servers.

C.3: What are the long-term benefits of the MySQL Enterprise Monitor?

Over time, the task of managing even medium-scale MySQL server farms becomes exponentially more complicated, especially as the load of users, connections, application queries, and objects on each
MySQL server increases. The Enterprise Monitor continually monitors the dynamic security, performance, replication and schema relevant metrics of all MySQL servers, so as the number of MySQL continues to grow, DBAs are kept up to date on potential problems and proactive measures that can be implemented to ensure each server continues to operate at the highest levels of security, performance and reliability.

C.4: How is the MySQL Enterprise Monitor installed and deployed?

The Enterprise Monitor is powered by a distributed web application that is installed and deployed within the confines of the corporate firewall.

C.5: How is the Enterprise Monitor web application architected?

The Enterprise Monitor web application comprises three components:

- Monitor Agent: A lightweight Java program that is installed on each of the monitored hosts. Its purpose is to collect MySQL SQL and operating system metrics that allow the DBA to monitor the overall health, availability and performance of the MySQL server and host. The Monitor Agent is the only component within the application that touches or connects to the MySQL Server. It reports the data it collects via XML over HTTP to the centralized Service Manager.

- Service Manager: The main server of the application. The Service Manager manages and stores the data collections that come in from each monitor agent. It analyzes these collections using MySQL provided best practice Advisor rules to determine the health, security, availability and performance of each of the monitored MySQL Servers. The Service Manager also provides the content for the Enterprise User Interface which serves as the client user interface for the distributed web application.

- Repository: A MySQL database that is used to stored data collections and application-level configuration data.

C.6: What makes MySQL Enterprise unique?

Of the products on the market that monitor MySQL, SQL code and OS specific metrics, the MySQL Enterprise Monitor is the only solution that is built and supported by the engineers at MySQL. Unlike other solutions that report on raw MySQL and OS level metrics, the MySQL Enterprise Monitor is designed to optimize the use of MySQL by proactively monitoring MySQL instances and providing notifications and 'MySQL DBA expertise in a box' advice on corrective measures DBAs can take before problems occur.

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

The MySQL Enterprise Monitor supports MySQL versions 5.1 and above.

C.8: What operating system platforms are supported by the MySQL Enterprise Monitor?

The Enterprise Monitor Service Manager is fully supported on most current versions of Linux, Windows and Windows Server Editions, and Solaris. The Monitor Agent supports any platform supported by the MySQL Enterprise server. For the complete list of MySQL Enterprise supported operating systems and CPUs, visit MySQL Supported Platforms and select MySQL Enteprise Monitor.

C.9: How are subscribers notified about the availability of new or updated MySQL Enterprise Monitor, MySQL Enterprise Advisors and Advisor Rules?

Customers receive email notifications of new and updated MySQL Enterprise Monitor versions. Also, the What's New section of MySQL Enterprise Monitor, if enabled, contains new product announcements.
• 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 29.1, “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 handle prepared statements?

At this time, the query analyzer does not track server-side prepared statements. However the default configurations for most client-side libraries for MySQL don't use them, they emulate them client-side, and those will be tracked by the query analyzer.

C.11: How much degradation in performance does mysql-proxy introduce?

At the very least it's equivalent to a network hop in latency. The degradation is directly related to your average query execution time. If your queries execute in microseconds (which can happen if served from query cache) then the degradation will be higher, and noticeable. We've seen some applications that actually do work when they execute queries, the degradation is much less, and in some limited cases because of scheduling, the application actually has better throughput.

C.12: Will the Query Analyzer work without any special setup?

With MySQL Server 5.6.14 and greater, Query Analyzer data is automatically (by default) collected and displayed using the Performance Schema Statement Digests MySQL Server feature. If you are monitoring an earlier MySQL Server version, then you can continue to use alternative methods of providing query data to the Query Analyzer.

For information about the different methods of retrieving query data, see Section 29.1, “Providing Query Analyzer Data”. 
This appendix describes the Diagnostics Report.

D.1 Diagnostics Report

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

Introduction

To generate a diagnostic report file, select Diagnostics Report from the Settings menu. The information is provided as a time stamped Zip file (such as support-20160115T2238.zip) that is downloaded to the machine. The information contained in the report includes detailed information about your server (or multiple servers if you selected a server group), including configuration, hardware, MySQL options/variables and historical graphs. To view the information extracted, unzip the downloaded file and double-click the index.html.

This report is very useful for debugging the MySQL Enterprise Service Manager and the MySQL Enterprise Monitor Agent. When filling out a My Oracle Support (MOS) ticket include this report.

Diagnostics Report File Contents

- **audit.log**: The Audit log file.
- **catalina.out**: A Tomcat log file.
- **java-threads.dot**: A list of the current Java threads and the dependencies.
- **java.props**: The current Java configuration properties.
- **java.threads**: A list of the current Java threads and their backtrace.
- **mysql-monitor.log**: The general MySQL Enterprise Service Manager log file.
- **mysql-monitor-full.log**: The full MySQL Enterprise Service Manager log file, that also contains stack traces.
- **mysql-monitor-agent-full.log**: A full built-in MySQL Enterprise Monitor Agent log file, that also contains stack traces.
- **preferences.properties**: The MySQL Enterprise Monitor preference settings.
- **product usage.html**: A usage report for each MySQL Enterprise Monitor User Interface page.
• **query.instanceOverview.html**: An HTML list of the current query instance related information.

  ![Note]
  The format of this file changed in 3.0. It is now listed as one Asset per block, instead of having one row per Asset inventory item.

• **Replication 1.json**: The calculated MySQL server replication structure.

• **root.csv**: A copy of your main MySQL Enterprise Monitor log file.

• **server.props**: A copy of your server properties.

• **tomcat.log**: A Tomcat log file.
MySQL Enterprise Monitor Glossary

These terms are commonly used in information about the MySQL Enterprise Monitor product.

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 25, 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.

Aggregator
An optional component, separately installed, which can intercept query data as it passes between applications and the database, and transmit that data to the Service Manager. To collect query data from applications requires using a communications chassis, such as MySQL Enterprise Agent Proxy Service or the appropriate Connector based on the application language or framework.

See Chapter 11, Proxy and Aggregator Installation for more information.
See Also Agent, Connector, plugin, 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 Dashboard, Service Manager, Tomcat.

asset
A component that the MySQL Enterprise Monitor product monitors, such as a MySQL server instance, or a CPU or file system within a server machine. Contrast with metric, which is a property within the component that is measured.
See Also instance, metric.

Asset Selector
The Groups shown in the Asset Selector are the ones you create and manage (on the MySQL Instances Dashboard, or during Agent installation) and those that are discovered automatically from monitoring replication topologies. It is on the left side of many UI pages that controls content in the main-body of the page.
See Also asset, Dashboard, graph.

autodiscovery
The mechanism that detects the set of master, slave, and master/slave servers in a replication configuration, and displays the topology in the Server tree.
See Also Asset Selector, master, replication, slave, topology.

B
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
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.

Connector
A software component (analogous to a “driver”) that provides connectivity to the MySQL server for client programs. MySQL comes with connectors for several programming languages and frameworks. The MySQL Enterprise Monitor product includes plugins for certain collectors, to help monitor queries sent by application programs and use that data within the Query Analyzer.
See Also Aggregator, Connector/J, Connector/NET, Connector/PHP, query, Query Analyzer.

Connector/J
The Connector for applications written in the Java programming language. It communicates directly with the MySQL server without going through the libmysqlclient shared library.

See Also Connector, Java, Query Analyzer.

Connector/NET
The Connector for applications written with the .NET programming framework. It communicates directly with the MySQL server without going through the libmysqlclient shared library.

See Section 12.3, “Using the MySQL Enterprise Plugin for Connector/NET” for more information.
See Also .NET, Connector, Query Analyzer.

Connector/PHP
The Connector for applications written in the PHP programming language. The mechanism for communicating with the MySQL server may differ depending on the version of PHP involved.

See Section 12.1, “Using the MySQL Enterprise Plugin for Connector/PHP” for more information.
See Also Aggregator, Connector, PHP, Query Analyzer.

credentials
The MySQL Enterprise Monitor product interfaces with many different databases and other kinds of servers. Each of these components can have its own login and security credentials. MySQL Enterprise Monitor pulls performance data from the MySQL servers that you monitor, stores the resulting data in a repository that is also a MySQL server, and sends alerts by communicating through other kinds of servers such as SMTP for e-mail alerts and NMS for SNMP traps. It pulls support-related data from the My Oracle Support site. You view the results in the UI, which is protected by its own login and optionally by LDAP authentication.
See Also Dashboard, LDAP, NMS, repository, SMTP, SNMP.

Critical
A high severity level for alarms. Within a rule, you can designate the threshold values that specify whether an alarm triggers a Critical, Warning, or notice alert.
See Also advisor, event, Notice, threshold, Warning.

custom
User-created rules and graphs. Contrast with standard.
See Also graph, standard.

D

Dashboard
The Dashboard menu contains several pages (dashboards) for viewing and configuring MySQL Enterprise Monitor. This includes dashboards for the Overview, Replication, and MySQL Instances.
See Also Events, Query Analyzer view, Service Manager, Settings, Tomcat, What's New tab.

data collection item
Data values derived from server status variables, operating system status information, and MySQL table information. You can reference these items using mnemonic names in expressions when you create or edit rules.
See Also expression.

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 22.1, “Events”.
See Also Dashboard, 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. These features are language-dependent, relying on support in the various Connectors for programming languages and frameworks.
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 monitor graphs using the Reports and Graphs page.
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 kinds of asset that the MySQL Enterprise Monitor product can monitor.

The MySQL Instances dashboard manages the instances, and the MySQL Process Discovery advisor discovers local instances on the host that the Agent is installed on.
See Also asset.

instrument
A persisted instance of a run-time metric evaluation. These may store the raw metric data, or the result of an expression or function against a metric. Instruments are generally stored for things that will show in the GUI, such as graph data.
See Also graph, metric.

inventory
Core information about a host or MySQL server instance. This data is collected by the Agent. The inventory includes details such as the MySQL server version number, supported storage engines and replication configuration. The data in the inventory helps to determine what other kinds of data can be collected from the hosts and MySQL servers.
See Also Agent, instance, replication.

J

Java
The MySQL Enterprise Plugin for Connector/J enables any application using the Connector/J JDBC driver to supply query analyzer information directly to MySQL Enterprise Service Manager. Information is sent to the MySQL Enterprise Service Manager for analysis without any need to modify your applications.

The MySQL Enterprise Monitor product is 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 Connector/J, Dashboard, JDBC, plugin, Query Analyzer, Tomcat.

JDBC
The underlying database API used by many Java applications, including the Connector/J component.
See Also Connector/J, Java.

L
LDAP
An authentication mechanism that can control access to the UI. On Linux, Unix, and OS X systems, you might have a separate LDAP server where each user has their own credentials. On Windows systems, the LDAP protocol is used to connect to an Active Directory server for the same authentication purposes. Typically, you map LDAP roles to corresponding MySQL Enterprise Monitor roles, to enable groups of users to have basic or administrative access to the Dashboard without configuring each user individually, or giving them a new user ID and password, or requiring extra work to revoke access when they leave the organization.
See Also credentials, role.

clog 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 Dashboard, repository, Service Manager, Tomcat.

Lua
A programming language that is used for parts of the Aggregator and Service Manager. Although you do not need to know this language to operate MySQL Enterprise Monitor, you may need to specify options containing file paths related to Lua scripts or libraries.
See Also Agent, Aggregator, 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.
monitor
To view information about the state, health, activity, and history of a resource such as a MySQL server instance. Monitoring can help to diagnose problems, spot worrisome trends before they turn into problems, reassure when systems are operating normally, and notify when an operator needs to take corrective action. With MySQL Enterprise Monitor, the component you interact with during monitoring is the UI.
See Also Dashboard, instance.

My Oracle Support
The 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, including the Connector/NET component.
See Also Connector/NET.

NMS
Acronym for Network Management System, a type of monitoring system that is separate from the MySQL Enterprise Monitor product. If your network has this kind of monitoring capability, MySQL Enterprise Monitor can notify the NMS of network issues by translating events into SNMP traps.
See Also event, SNMP.

normalized query
A condensed form of the query text used to treat similar queries as if they were identical, for monitoring performance. When MySQL Enterprise Monitor normalizes queries, it disregards differences in keyword capitalization, whitespace, and most comments. It replaces literal values with placeholders and transforms multi-row insert statements and IN clauses, to group similar statements with different parameters when measuring how much time is consumed by a particular type of query.
See Also canonical query, query.

Notice
A low severity level for events. Within an advisor, you can designate the threshold values that specify the severity level that is triggered.
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.
PHP
A scripting language typically embedded within web pages.
See Also Connector/PHP.

plugin
In MySQL Enterprise Monitor, a component that sits between the database and applications that use a particular language-based Connector. It collects query data from both standalone and web-based applications, for MySQL Enterprise Monitor to monitor. For example, there is the MySQL Enterprise Plugin for Connector/NET, and the MySQL Enterprise Plugin for Connector/J. The mechanism for transferring the data, such as whether the data flows through the Aggregator or the Proxy, depends on the particular connector.
See Also Aggregator, Connector, proxy.

proxy
An optional, and separately installed, component. It can accept queries from user applications, forward them to a MySQL server, and return the query results back to the application. You rearrange the TCP/IP ports so that the application communicates with the proxy instead of the database, either by changing the port where the application communicates with the database, or the port where the database listens.
See Also Agent, Aggregator.

Q

QRTi
QRTi stands for “Query Response Time index”. It is a "quality of service" measurement for each query, and it uses the Apdex formula for that calculation. For more information about Apdex, see: Apdex on Wikipedia.
See Also Query Analyzer.

QuAn
An abbreviation for Query Analyzer used in some keyword names.
See Also Query Analyzer.

query
In the MySQL Enterprise Monitor context, any SQL statement whose performance might be monitored. Includes not just SQL SELECT queries, but also DML statements such as INSERT, UPDATE, and DELETE, and DDL statements such as CREATE TABLE and TRUNCATE TABLE.
See Also normalized query, slow query log.

Query Analyzer
The MySQL Enterprise Monitor component that tracks data about MySQL queries and summarizes that data using graphs and tables. You interact with it (for example, filtering the displayed queries or selecting a specific time period) using the Query Analyzer tab in the UI.
See Also Dashboard, graph, query, 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 29, Using the Query Analyzer 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 dashboard in the UI.

See Also Dashboard, Replication dashboard, topology.

Replication dashboard

A dashboard page within the UI that monitors aspects of replication. See Chapter 19, Replication Dashboard.

See Also Dashboard, replication.

repository

The database that stores the monitoring data collected by the MySQL Enterprise Monitor product. It can be a separate database instance that is part of the MySQL Enterprise Monitor installation (the bundled MySQL server), or you can use an existing database of your own. The bundled MySQL server is a level of MySQL database that is fully tested with the MySQL Enterprise Monitor product, and can be kept separate from your other databases to avoid any extra load on them. You might use an existing server that has spare capacity, fast storage devices, tuned configuration parameters, a backup system, or other conveniences that can benefit the MySQL Enterprise Monitor data storage as well.

See Also bundled, instance.

ring replication

See circular replication.

role

A level of access privilege for the UI. One of manager (highest privilege), dba, read-only, or agent (specialized privilege for sending data). Each user account registered with the Dashboard must have one of these roles. Components such as the Agent and the Aggregator use the agent role. To simplify the process of granting credentials for large numbers of users, you can map LDAP roles to these MySQL Enterprise Monitor roles.

See Also Agent, Aggregator, credentials, Dashboard, LDAP.

S

schedule

Determines how often each advisor is evaluated.

See Also advisor.

scope

A dynamic filter against an asset class.
Service Manager
The core MySQL Enterprise Monitor component that receives the monitoring data from the Agent, Aggregator, and Query Analyzer components. It displays this information through the GUI web-interface, and manages the Advisors, Events, and Event Handlers, as well as access to Reporting, Graphing, and Query Analysis.
See Also advisor, Agent, Aggregator, 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, slave.

topology view
A graphical representation of the replication topology displayed in the Replication dashboard.

See Also replica, replication, 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, Dashboard, master/slave, replication.

trending
Predicting when a problem may occur. For example, predicting that a disk will run out of disk space 30 days in the future.

U

UI
The UI is a web-based interface to the MySQL Enterprise Service Manager. The back end is a Java application powered by the Tomcat server.
See Also Dashboard.

UUID
A unique identifier used to distinguish each MySQL instance, host machine, and agent. Because there is so much flexibility in spreading components across multiple systems or running multiple instances and agents on the same system, the combination of these different UUIDs identifies where information came from and the source of any issues. Always generate a new UUID for any one of these components, rather than copying or reusing an existing UUID value.

In MySQL 5.5, or earlier, the UUID for the server is stored in the table mysql.inventory. When a component such as the Agent or a Connector plugin connects to that MySQL server, the applicable MySQL user must have privileges to read this table.
See Also Agent, instance.

W

Warning
A medium severity level for alarms. Within a rule, you can designate the threshold values that specify for each severity level.
See Also advisor, 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 Dashboard, My Oracle Support.
Symbols
.NET connector plug-in, 110
, 83

A
access control, 329
Administration Advisors, 199
  32-Bit Binary Running on 64-Bit AMD Or Intel System, 200
  Binary Log Debug Information Disabled, 200
  Binary Logging Is Limited, 200
  Binary Logging Not Enabled, 200
  Binary Logging Not Synchronized To Disk At Each Write, 201
  Binary Logs Automatically Removed Too Quickly, 201
  Database May Not Be Portable Due To Identifier Case Sensitivity, 201
  Event Scheduler Disabled, 201
  General Query Log Enabled, 202
  Host Cache Size Not Sufficient, 202
  In-Memory Temporary Table Size Limited By Maximum Heap Table Size, 202
  InnoDB Status Truncation Detected, 203
  InnoDB Strict Mode Is Off, 203
  InnoDB Tablespace Cannot Automatically Expand, 203
  InnoDB Transaction Logs Not Sized Correctly, 203
  Multiple Threads Used When Repairing MyISAM Tables, 204
  MySQL Server No Longer Eligible For Oracle Premier Support, 204
  Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 204
  No Value Set For MyISAM Recover Options, 204
  Table Cache Set Too Low For Startup, 204
  Time Zone Data Not Loaded, 205
  Warnings Not Being Logged, 205
--adminpassword option, 67
--adminuser option, 67
advisor, 329
advisor category, 329
Advisors, 235
  32-Bit Binary Running on 64-Bit AMD Or Intel System, 200
  Account Has An Overly Broad Host Specifier, 249
  Account Has Global Privileges, 249
  Account Has Old Insecure Password Hash, 228
  Account Requires Unavailable Authentication Plug-ins, 228
  Agent Health Advisor, 235
Attempted Connections To The Server Have Failed, 206
AUTO_INCREMENT Field Limit Nearly Reached, 223
Average Statement Execution Time Advisor, 244
Binary Log Checksums Disabled, 218
Binary Log Debug Information Disabled, 200
Binary Log File Count Exceeds Specified Limit, 219
Binary Log Row Based Images Excessive, 219
Binary Log Space Exceeds Specified Limit, 219
Binary Log Usage Exceeding Disk Cache Memory Limits, 213
Binary Logging Is Limited, 200
Binary Logging Not Enabled, 200
Binary Logging Not Synchronized To Disk At Each Write, 201
Binary Logs Automatically Removed Too Quickly, 201
Cluster Data Node Data Memory Getting Low, 208
Cluster Data Node Has Been Restarted, 208
Cluster Data Node Index Memory Getting Low, 208
Cluster Data Node Redo Buffer Space Getting Low, 208
Cluster Data Node Redo Log Space Getting Low, 208
Cluster Data Node Undo Buffer Space Getting Low, 208
Cluster Data Node Undo Log Space Getting Low, 208
Cluster Data Nodes Not Running, 209
Cluster DiskPageBuffer Hit Ratio Is Low, 209
Cluster Has Stopped, 209
CPU Utilization Advisor, 241
Database May Not Be Portable Due To Identifier Case Sensitivity, 201
Database-level Privileges on All Databases, 249
Duplicate Host Identity, 240
Duplicate MySQL Server UUID, 240
Event Scheduler Disabled, 201
Excessive Disk Temporary Table Usage Detected, 213
Excessive Number of Locked Processes, 214
Excessive Number of Long Running Processes, 214
Excessive Number of Long Running Processes Locked, 214
Excessive Percentage Of Attempted Connections To The Server Have Failed, 206
Filesystem Free Space Advisor, 242
Flush Time Set To Non-Zero Value, 214
General Query Log Enabled, 202
Group Replication, 246
Group Replication Configuration, 246
Group Replication Status, 247
Host Cache Size Not Sufficient, 202
HTTP Server Performance, 211
In-Memory Temporary Table Size Limited By Maximum Heap Table Size, 202
Indexes Not Being Used Efficiently, 214
InnoDB Buffer Cache Has Sub-Optimal Hit Rate, 209

341
InnoDB Buffer Pool Writes May Be Performance Bottleneck, 215
InnoDB Flush Method May Not Be Optimal, 215
InnoDB Log Buffer Flushed To Disk After Each Transaction, 215
InnoDB Log Waits May Be Performance Bottleneck, 215
InnoDB Not Using Newest File Format, 215
InnoDB Status Truncation Detected, 203
InnoDB Strict Mode Is Off, 203
InnoDB Tablespace Cannot Automatically Expand, 203
InnoDB Transaction Logs Not Sized Correctly, 203
Insecure Password Authentication Option Is Enabled, 228
Insecure Password Generation Option Is Enabled, 228
Key Buffer Size May Not Be Optimal For Key Cache, 210
LOCAL Option Of LOAD DATA Statement Is Enabled, 228
Master Not Verifying Checksums When Reading From Binary Log, 220
Maximum Connection Limit Nearing Or Reached, 207
Multiple Threads Used When Repairing MyISAM Tables, 204
MyISAM Concurrent Insert Setting May Not Be Optimal, 216
MyISAM Indexes Found with No Statistics, 225
MySQL Agent Memory Usage Excessive, 205
MySQL Agent Not Reachable, 205
MySQL Availability, 207
MySQL Enterprise Backup Health Advisor, 238
MySQL Process Discovery Advisor, 239
MySQL Server Has Been Restarted, 207
MySQL Server No Longer Eligible For Oracle Premier Support, 204
MySQL User Account, 249
Next-Key Locking Disabled For InnoDB But Binary Logging Enabled, 204
No Value Set For MyISAM Recover Options, 204
Object Changed: Database Has Been Altered, 224
Object Changed: Database Has Been Created, 224
Object Changed: Database Has Been Dropped, 224
Object Changed: Function Has Been Created, 224
Object Changed: Function Has Been Dropped, 224
Object Changed: Index Has Been Created, 225
Object Changed: Index Has Been Dropped, 225
Object Changed: Table Has Been Altered, 226
Object Changed: Table Has Been Created, 226
Object Changed: Table Has Been Dropped, 226
Object Changed: User Has Been Dropped, 227
Object Changes Detected, 225
Policy-Based Password Validation Does Not Perform Dictionary Checks, 229
Policy-Based Password Validation Is Weak, 229
Policy-Based Password Validation Not Enabled, 229
Prepared Statements Not Being Closed, 216
Prepared Statements Not Being Used Effectively, 216
Privilege Alterations Detected: Privileges Granted, 230
Privilege Alterations Detected: Privileges Revoked, 230
Privilege Alterations Have Been Detected, 230
Query Analysis Reporting, 245
Query Cache Has Sub-Optimal Hit Rate, 210
Query Cache Is Excessively Fragmented, 216
Query Cache Potentially Undersized, 210
Query Pileup Advisor, 244, 245
Replica Execution Position Too Far Behind, 220
Replication Configuration Advisor, 220
Replication Status Advisor, 220
Replication Too Far Behind, 221
Root Account Can Login Remotely, 230
Root Account Without Password, 230
Server Contains Default "test" Database, 231
Server Has Accounts Without A Password, 231
Server Has Anonymous Accounts Not Being Used As Proxies, 231
Server Has No Locally Authenticated Root User, 231
Server Includes A Root User Account, 232
Server-Enforced Data Integrity Checking Disabled, 225
Server-Enforced Data Integrity Checking Not Strict, 226
SHA-256 Password Authentication Not Enabled, 231
Slave Detection Of Network Outages Too High, 220
Slave Has Login Accounts With Inappropriate Privileges, 221
Slave Not Configured As Read Only, 221
Slave Not Verifying Checksums When Reading From Relay Log, 221
Slave Relay Log Space Is Very Large, 222, 222
Slave SQL Processing Not Multi-Threaded, 222
Slave Without REPLICATION SLAVE Accounts, 222
SQL Statement Generates Warnings or Errors, 245
Syslinks Are Enabled, 232
sys Schema Install Advisor, 241
Table Cache Not Optimal, 210
Table Cache Set Too Low For Startup, 204
Table Lock Contention Excessive, 217
Tables Found with No Primary or Unique Keys, 226
Thread Cache Not Enabled, 217
Thread Cache Size May Not Be Optimal, 210
Thread Pool Stall Limit Too Low, 217
Thread Pooling Not Enabled, 217
Time Zone Data Not Loaded, 205
Too Many Concurrent Queries Running, 218
Unsupported Agent Version, 212
User Has Rights To Database That Does Not Exist, 232
User Has Rights To Table That Does Not Exist, 232
Users Can View All Databases On MySQL Server, 232
Warnings Not Being Logged, 205
advisors
creating, 277, 282
agent
installation, 33
troubleshooting, 46
Agent, 329
Agent Advisors, 205
Agent Health Advisor, 235
MySQL Agent Memory Usage Excessive, 205
MySQL Agent Not Reachable, 205
--agent_autocreate option, 74
--agent_installtype option, 76
--agentpassword option, 70
--agentservicename option, 76
--agentuser option, 71
Aggregator, 329
PHP connector, 101
Apache, 329
Apple OS X, 42, 115, 313
architecture, 5
asset, 329
Asset Selector, 329
autodiscovery, 330
Availability Advisors, 206
Attempted Connections To The Server Have Failed, 206
Excessive Percentage Of Attempted Connections To The Server Have Failed, 206
Maximum Connection Limit Nearing Or Reached, 206
MySQL Availability, 207
MySQL Server Has Been Restarted, 207

B
backlog, 330
backup
restore, 51
Backup Advisors
MySQL Enterprise Backup Health Advisor, 238
Backup Dashboard
Backup Details, 155
filter, 155
Group Overview, 153
History, 154
Instance Overview, 153
Backup Details
Backup Command, 157
Backup Configuration, 156
Backup Status, 156
InnoDB Configuration, 157
Progress Log, 157
--backupdir option, 63
backups, 58
blackout period, 288, 330
bottleneck, 330
browsers, 24
bundled, 330
C
C, 330
canonical query, 330
chain replication, 330
--checkmysqldhost option, 71
circular replication, 330
Cluster Advisors, 207
Cluster Data Node Data Memory Getting Low, 207
Cluster Data Node Has Been Restarted, 208
Cluster Data Node Index Memory Getting Low, 208
Cluster Data Node Redo Buffer Space Getting Low, 208
Cluster Data Node Redo Log Space Getting Low, 208
Cluster Data Node Undo Buffer Space Getting Low, 208
Cluster Data Node Undo Log Space Getting Low, 208
Cluster Data Nodes Not Running, 209
Cluster DiskPageBuffer Hit Ratio Is Low, 209
Cluster Has Stopped, 209
cfg.properties file, 314
configuration files, 313
configuration utilities
agent, 86
agent.bat, 86
agent.sh, 86
config.bat, 83
config.sh, 83
service manager, 83
Connector, 330
Connector/J, 105, 331
Connector/Net, 110
Connector/NET, 331
Connector/PHP, 331
--createBackup option, 70
--createDataBackup option, 64
credentials, 331
Critical, 331
custom, 331
custom data collection
customizing advisors, 284
D
Dashboard, 331
data collection item, 331
Query Cache Has Sub-Optimal Hit Rate, 210
Query Cache Potentially Undersized, 210
Table Cache Not Optimal, 210
Thread Cache Size May Not Be Optimal, 210
metric, 334
MIB file, 313
mixed replication, 334
--mode option, 64, 72
monitor, 335
installation, 25
Monitoring and Support Advisors
sys Schema Install Advisor, 241
Monitoring and Support Services Advisors, 211
Duplicate Host Identity, 240
Duplicate MySQL Server UUID, 240
HTTP Server Performance, 211
MySQL Process Discovery Advisor, 239
Support Diagnostics, 211
Unsupported Agent Version, 212
My Oracle Support, 335
MySQL Enterprise Backup, 58
MySQL Server
installation, 25
MySQL server, 335
mysql-monitor-agent.log file, 317
--mysqlconnectiongroup option, 76
--mysqlconnmethod option, 73
mysqldump, 58
mysqlenterprise.* options, 101, 105
--mysqlhost option, 73
--mysql-identity-source option, 66
--mysqlpassword option, 74
--mysqlport option, 74
--mysqlsocket option, 74
--mysqluser option, 74
--mysql_installation_type option, 68

N
Named Filters
Creating Filters, 129
.NET, 335
NMS, 335
normalized query, 335
Notice, 335
notification, 335
notification group, 335

O
Operating System Advisors, 212
CPU Utilization Advisor, 241
Filesystem Free Space Advisor, 242
--optionfile option, 64, 75
OS X, 42, 115, 313

P
Performance Advisors, 212
Binary Log Usage Exceeding Disk Cache Memory Limits, 213
Excessive Disk Temporary Table Usage Detected, 213
Excessive Number of Locked Processes, 213
Excessive Number of Long Running Processes, 214
Excessive Number of Long Running Processes Locked, 214
Flush Time Set To Non-Zero Value, 214
Indexes Not Being Used Efficiently, 214
InnoDB Buffer Pool Writes May Be Performance Bottleneck, 215
InnoDB Flush Method May Not Be Optimal, 215
InnoDB Log Buffer Flushed To Disk After Each Transaction, 215
InnoDB Log Waits May Be Performance Bottleneck, 215
InnoDB Not Using Newest File Format, 215
MyISAM Concurrent Insert Setting May Not Be Optimal, 216
Prepared Statements Not Being Closed, 216
Prepared Statements Not Being Used Effectively, 216
Query Cache Is Excessively Fragmented, 216
Table Lock Contention Excessive, 217
Thread Cache Not Enabled, 217
Thread Pool Stall Limit Too Low, 217
Thread Pooling Not Enabled, 217
Too Many Concurrent Queries Running, 218
performance schema, 296
performance tuning, 79
PHP, 336
PHP connector plugin
Aggregator, 101
plugin, 336
ports
MySQL server for monitoring, 74
MySQL server for repository, 68
service manager, 66, 72
Tomcat, 66
Tomcat SSL, 66
post-install tasks, 53
proxy, 336

Q
QRTi, 336
QuAn, 336
query, 336
Query Analysis Advisors, 244
Average Statement Execution Time Advisor, 244
Query Analysis Reporting, 245
Query Pileup Advisor, 245, 245
SQL Statement Generates Warnings or Errors, 245
Query Analyzer, 127, 295, 336
.NET connector, 110
Java connector, 105
PHP connector, 101
supplying query data, 296
Query Analyzer tab, 300
Query Analyzer view, 336

R
remote, 336
remote monitoring, 337
replica, 337
Replication, 159
replication, 337
Replication Advisors, 218
Binary Log Checksums Disabled, 218
Binary Log File Count Exceeds Specified Limit, 219
Binary Log Row Based Images Excessive, 219
Binary Log Space Exceeds Specified Limit, 219
Master Not Verifying Checksums When Reading From Binary Log, 220
Replica Execution Position Too Far Behind, 220
Replication Configuration Advisor, 219
Replication Status Advisor, 220
Replication Too Far Behind, 221
Slave Detection Of Network Outages Too High, 220
Slave Has Login Accounts With Inappropriate Privileges, 221
Slave Not Configured As Read Only, 221
Slave Not Verifying Checksums When Reading From Relast Log, 221
Slave Relay Log Space Is Very Large, 222, 222
Slave SQL Processing Not Multi-Threaded, 222
Slave Without REPLICATION SLAVE Accounts, 222
Replication dashboard, 337
Reports and Graphs, 127
repository, 337
--restartImmediately option, 70
restore
backup, 51
restoring
backup, 51
role, 337
rules
blackout periods, 288
creating, 277
variable substitution, 280

S
schedule, 337
Schema Advisors, 223
AUTO_INCREMENT Field Limit Nearly Reached, 223
MyISAM Indexes Found with No Statistics, 225
Object Changed: Database Has Been Altered, 224
Object Changed: Database Has Been Created, 224
Object Changed: Database Has Been Dropped, 224
Object Changed: Function Has Been Created, 224
Object Changed: Function Has Been Dropped, 224
Object Changed: Index Has Been Created, 224
Object Changed: Index Has Been Dropped, 225
Object Changed: Table Has Been Altered, 225
Object Changed: Table Has Been Created, 226
Object Changed: Table Has Been Dropped, 226
Object Changed: User Has Been Dropped, 227
Object Changes Detected, 225
Server-Enforced Data Integrity Checking Disabled, 225
Server-Enforced Data Integrity Checking Not Strict, 225
Tables Found with No Primary or Unique Keys, 226
scope, 337
Security Advisors, 227, 248
Account Has Old Insecure Password Hash, 228
Account Requires Unavailable Authentication Plug-ins, 228
Insecure Password Authentication Option Is Enabled, 228
Insecure Password Generation Option Is Enabled, 228
LOCAL Option Of LOAD DATA Statement Is Enabled, 228
MySQL Enterprise Audit Plugin, 248
MySQL User Account, 248
Policy-Based Password Validation Does Not Perform Dictionary Checks, 229
Policy-Based Password Validation Is Weak, 229
Policy-Based Password Validation Not Enabled, 229
Privilege Alterations Detected: Privileges Granted, 230
Privilege Alterations Detected: Privileges Revoked, 230
Privilege Alterations Have Been Detected, 230
Root Account Can Login Remotely, 230
Root Account Without Password, 230
Server Contains Default "test" Database, 231
Server Has Accounts Without A Password, 231
Server Has Anonymous Accounts Not Being Used As Proxies, 231
Server Has No Locally Authenticated Root User, 231
Server Includes A Root User Account, 232
SHA-256 Password Authentication Not Enabled, 231
Symlinks Are Enabled, 232
User Has Rights To Database That Does Not Exist, 232
User Has Rights To Table That Does Not Exist, 232
Users Can View All Databases On MySQL Server, 232

service manager
database name, 68
installation, 25
Service Manager, 338
service request, 338
services
starting and stopping, 29
Settings, 338
severity, 338
slave, 338
slow query log, 338
SMTP, 338
SNMP, 338
SNMP traps, 313
source, 338
source/replica, 338
sql_mode, 44
SSH tunneling, 45
SSL, 66, 339
standard, 339
starting
MySQL Enterprise Monitor service, 30
MySQL Enterprise Monitor services, 29
Status Summary, 339
Instance Status, 129
stopping
MySQL Enterprise Monitor service, 30
MySQL Enterprise Monitor services, 29
support files
diagnostic report, 327
support issue, 339
--system-size option, 66

T
Table Statistics Report, 176
threshold, 339
thresholds, 281
timezone, 126
Tomcat, 339
installation, 25
starting and stopping, 30
--tomcatport option, 66
--tomcatsslport option, 66
topology, 339
topology view, 339
tree, 339
trending, 340
troubleshooting, 46
tuning, 79

U
UI, 340
unattended installation, 61
--unattendedmodeui option, 67, 75
uninstalling, 113, 113
Unix, 43, 114, 313
User Menu
Logout, 128
User Preferences, 128
User Preferences
User Menu, 128
UUID, 340

V
variable substitution, 280
--version option, 63, 76

W
Warning, 340
Web browsers, 24
What's New, 131
What's New tab, 340
Wiki markup, 282
Windows, 30, 41, 113, 313