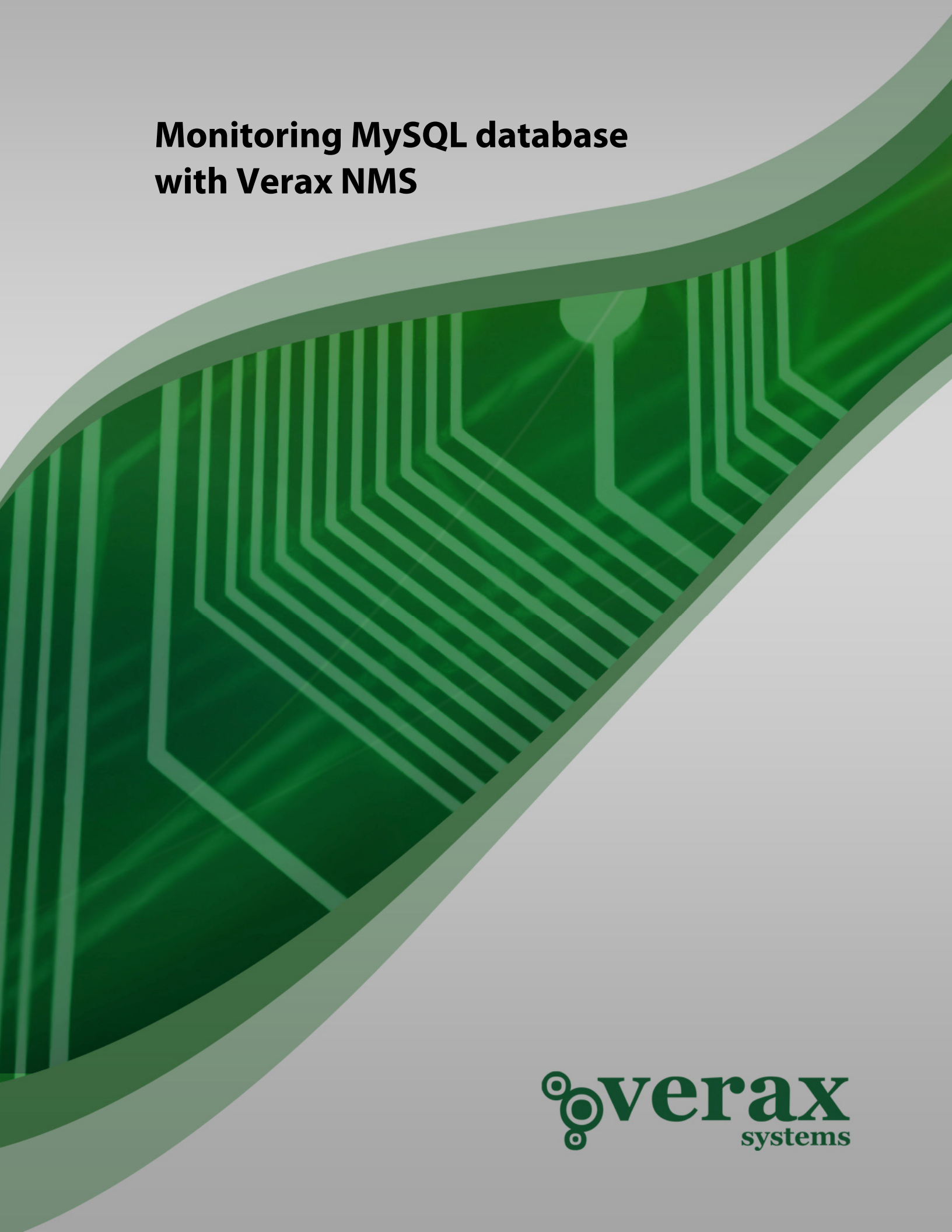
Monitoring MySQL database with Verax NMS



verax systems

Table of contents

Abstract

This publication provides an overview on how to monitor and manage MySQL database using Verax NMS (including the free Express edition available at http://www.veraxsystems.com/en/downloads - please read terms & conditions for limitations of the Express version).

Tools used:

- **MySQL Database:** http://www.mysql.com
- **Monitoring tool:** www.veraxsystems.com/en/products/nms

Agenda:

1. Adding MySQL database to the list of monitored applications within Verax NMS.
2. Configuring availability sensors and performance counters for the database.
3. MySQL database plugin features overview.
4. Setting up alarms and notification policies.

1. Adding MySQL database to device inventory

In order to include MySQL database instance to be monitored by Verax NMS, add an application instance to the device actually running this instance.

Note: Verax NMS allows for creating multiple instances for applications of the same type on a single device.

In order to add MySQL database server to the device running its instance, perform the following steps:

1. Log in into the Verax NMS and select *Home* from the main menu.
2. Select a device running MySQL database instance from the left-side *aspects* view.

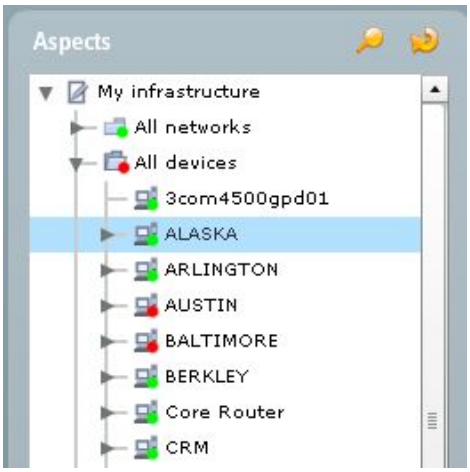


Figure 1: Aspect hierarchy tree

3. In *Summary tab* select ***Manage applications*** from the *actions* section.

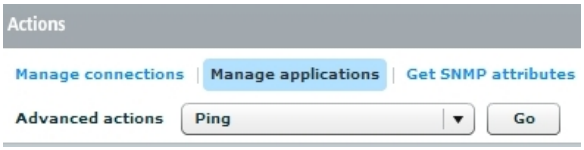


Figure 2: Adding a new managed application

4. A pop-up dialog is displayed.

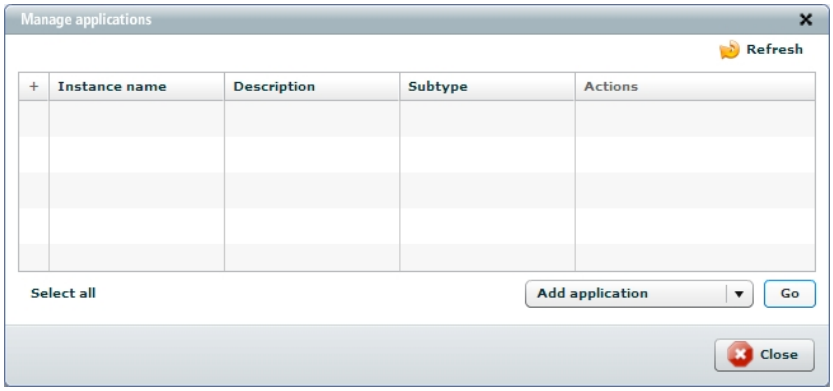


Figure 3: Applications list dialog

5. Select ***Add application*** option from the context menu and click ***Go***. A dialog window is displayed.

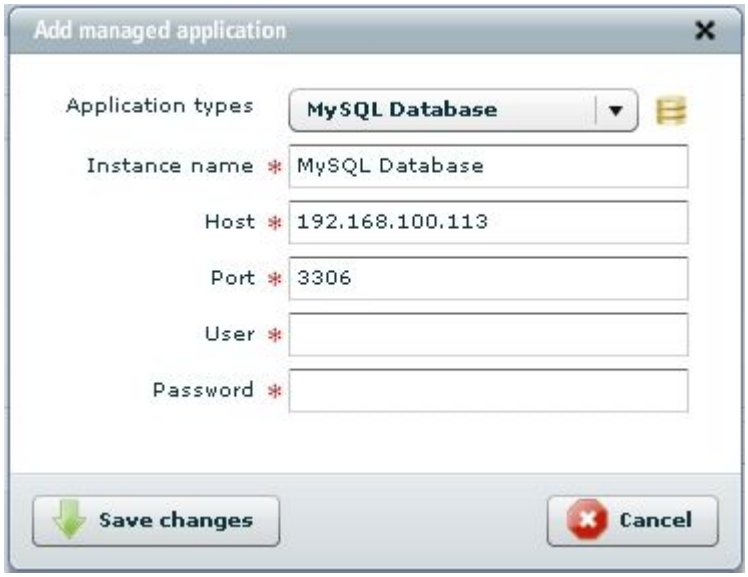


Figure 4: Adding MySQL database – parameters dialog

Verax NMS will ask to enter the following application-specific parameters:

- **Instance name** - Name of the application instance. You can enter any name describing the monitored application instance.
- **Host** - Address of the host running the application instance. In most cases, the host address is an IP address of the device the application instance is assigned to.
- **Port** - Port of application server (3306 by default).
- **User** - Username used to connect to the database (with administrative privileges).
- **Password** - Password used to connect to the database.

Note: application-specific parameters depend on the selected application type.

6. Provide the necessary information and click ***Save changes***.

7. The system will ask if you want to add a default set of sensors and counters for MySQL database. Since, in this example sensors and performance will be added manually – click ***No***.

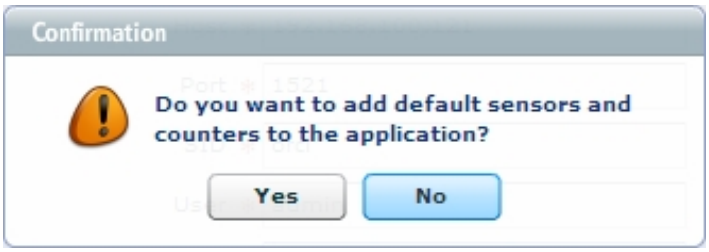


Figure 5: Default monitors - confirmation dialog

The initial set of MySQL database monitors includes:

- Configuration: software version, host platform, status and system variables
- Database instance inventory
- Predefined sensor and counter templates to monitor most important SQL performance characteristics.

8. The newly added database is now visible in the aspect tree within the host's node in ***Managed Applications*** category.

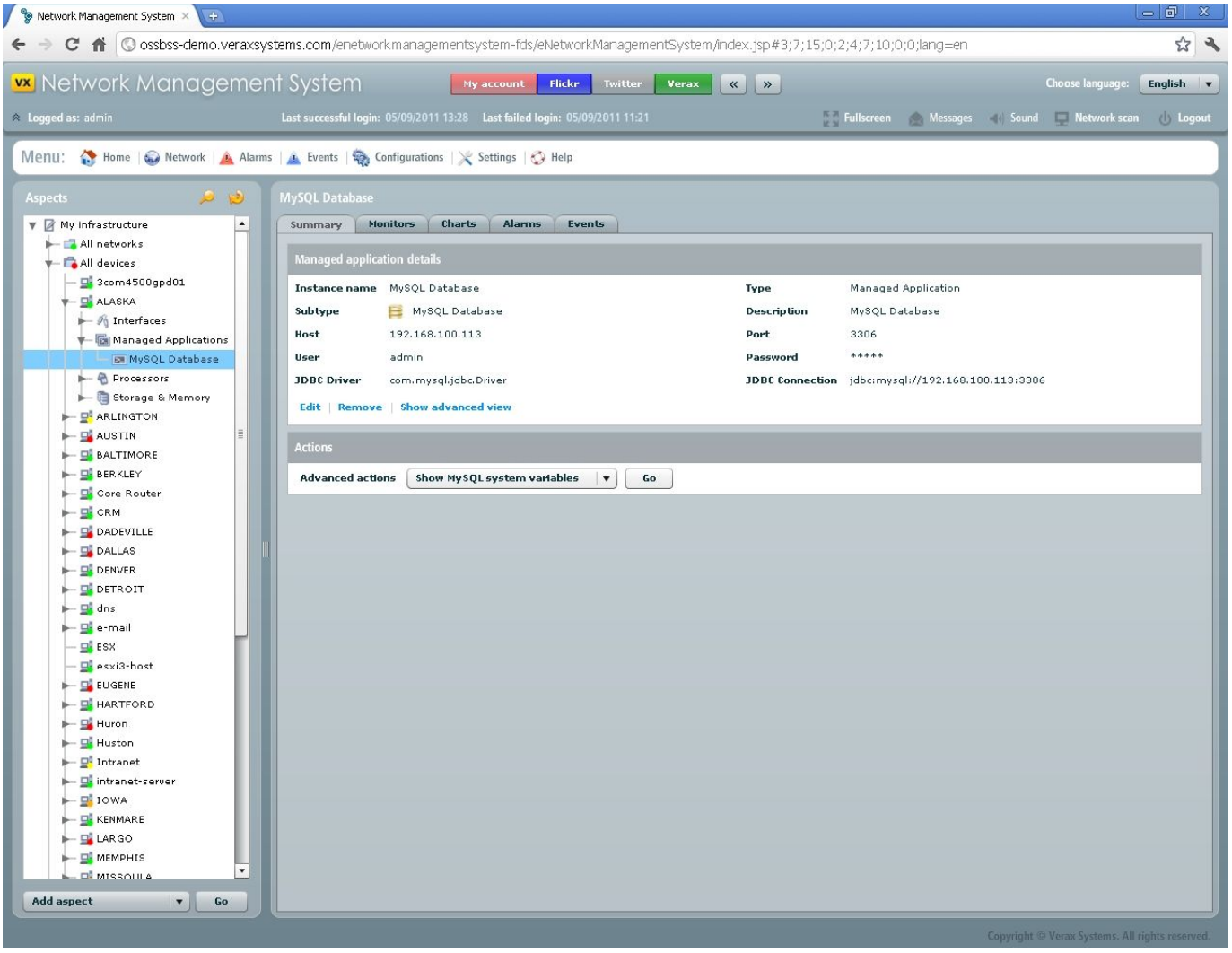


Figure 6: MySQL database properties window

2. Adding sensors for MySQL database

MySQL database instance can be monitored by:

- Obtaining database information and statistics on demand using the ***Show advanced view*** action available on ***Summary tab,*** which shows several views with major information about the database server as well as performance characteristics.
- Checking application or service availability by configuring SQL-based JDBC sensor.
- User-defined SQL-based JDBC counter.

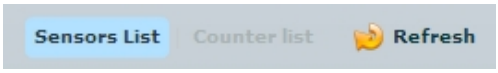
Sensors are active monitors periodically querying the device services for which they are configured and waiting for their responses. If a query is returned with an expected response, the queried service is considered "available." If a response is not received (timed out), or if the response is not as expected, the queried service is considered "unavailable".

The system includes one pre-configured sensor for MySQL database available by default:

- **JDBC** - checks if a given SQL query run on a specified database returns the expected result.

In order to add a sensor, perform the following steps:

1. Select device from the aspect tree in *Home view* (MySQL database in this case).
2. Select ***Monitors tab*** and switch to sensor list by clicking ***Sensor list*** link in the upper-right corner of the tab field. The sensor list is displayed.



3. Select ***Add*** from the global action menu and click ***Go***. The wizard dialog is displayed.

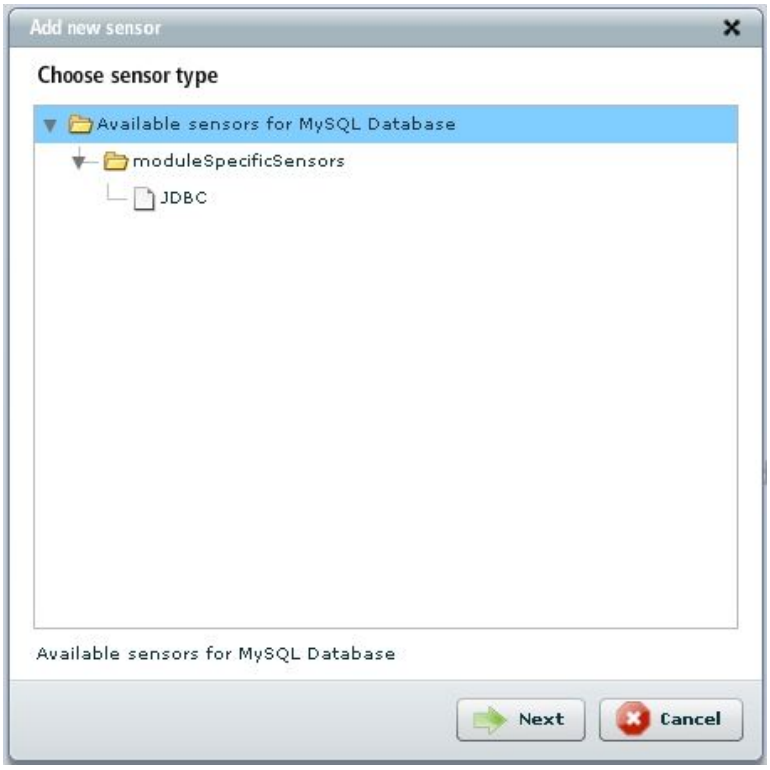


Figure 7: Adding a sensor

4. Select the sensor you want to add and click ***Next***.

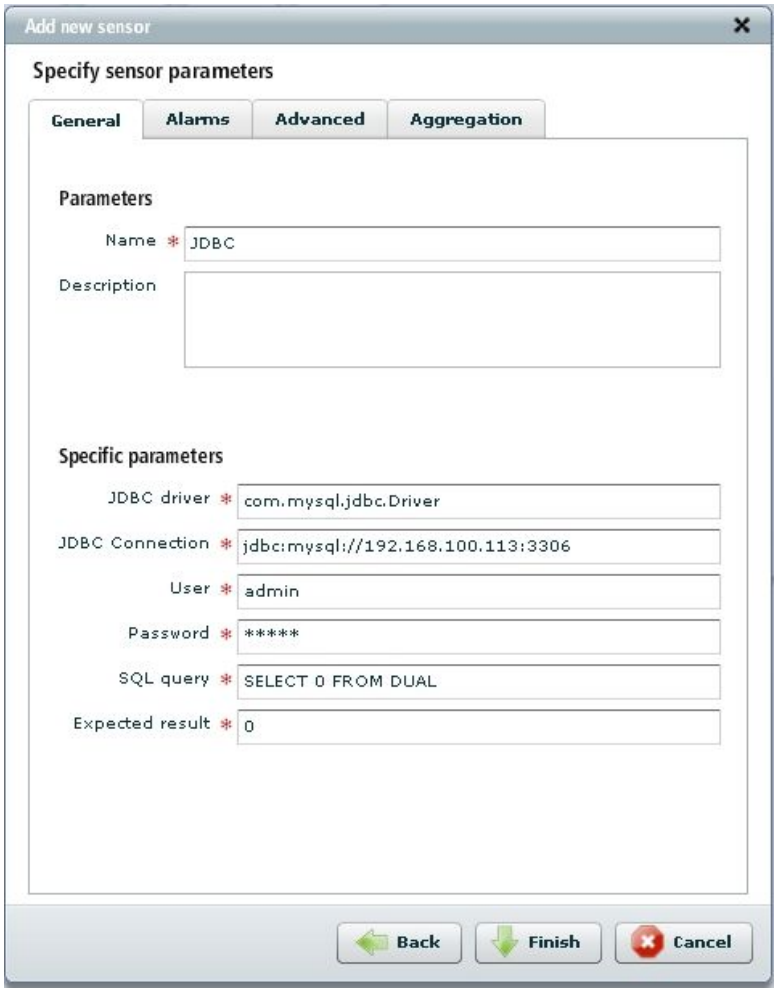


Figure 8: Specify sensor parameters

5. A dialog shows up with all sensor parameters to be provided. Specify the sensor parameters and click ***Finish***.
6. Once the sensors have been added, they are visible on the sensor list (*Monitors tab*).

3. Adding performance counters for MySQL database

Performance counters measure system activity and performance (metrics). The application retrieves their current values in predefined intervals. The aim of probing and collecting data is to analyze and convert the data into a performance graph/chart. The user can define a counter manually or load it from a template.

Performance counter templates are templates with defined probing parameters for specified devices in order to improve and speed up counter creation.

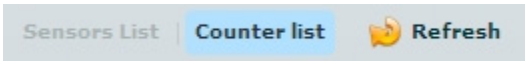
Each Verax NMS counter template is characterized by the following information:

- **Name and description** - unique identifier and optional description,
- **Device type** - type of a device,
- **Protocol type** - protocol used,
- **Probing interval** - pauses between probing.

Counter templates are needed when the counter creation method is set to "from template". Counter templates provide defined probing parameters for specified devices in order to improve and quicken counter creation.

In order to add user-defined SQL-based JDBC counter, perform the following steps:

1. Select device from aspect tree in *Home view*.
2. Select ***Monitors tab*** and switch to the counter list by clicking ***Counter list*** link in the upper-right corner of the tab field. A counter list is displayed.



3. Select ***Add*** from the global action menu and click ***Go***. Select the counter you want to create and click ***Next***.

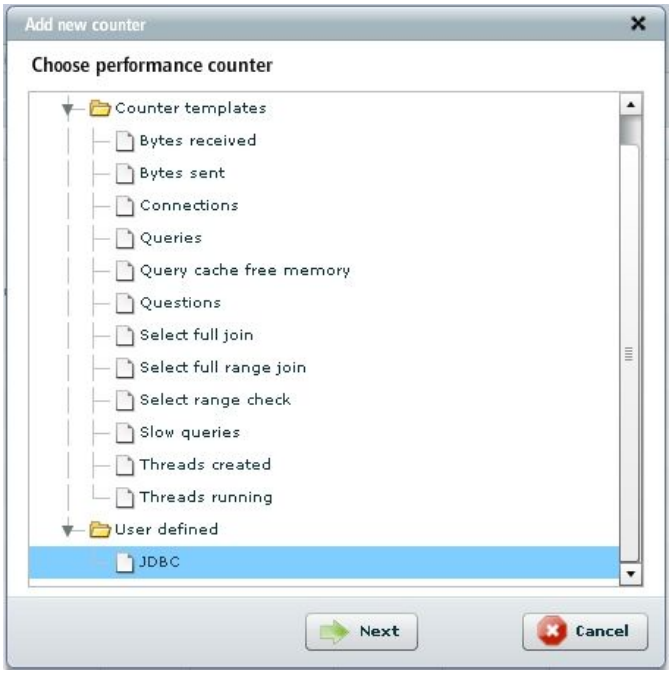


Figure 9: Adding counter for MySQL database

4. Once the data has been loaded, the edit window shows up with all counter attributes to be provided.

Figure 10: Selecting measure metrics

5. Specify the rest of the counter parameters (alarms, thresholds etc.) and click ***Finish***.
6. The new counter has been created and is now visible on the counter list.

4. Creating custom event processing rules for MySQL database

Events are processed by Event Processing Rules running under control of the Event Processing Engine. The Event Processing Engine within the system is able to process events fast without materializing them in database. Verax NMS comes with a set of embedded, flexibly customized processing rules such as: De-duplication, Pairwise matching, Event forwarding, Intermittent failure, Scheduled Maintenance, etc. It also provides users with the ability to implement their own processing rules using JRuby scripting language.

Verax NMS provides complex fault management, such as alarm collection, filtering, blocking, thresholds and correlation (scripted, user-defined rules defining business logic for alarm correlation, cleaning, root-cause, etc.) as well as alarm management actions, e.g. assignment, change of status, clearing, annotation and others. It also enables users to create alarms based on network data etc.

In this example it is shown how to assign basic event processing rules such as:

- Alarm generating
- Event dropping
- Event forwarding
- Severity assigning

To assign an event processing rule, perform the following steps:

1. After selecting the desired host go to *Events tab*.
2. Select events, choose ***Assign processing rules*** and click ***Go***.
3. A dialog window is displayed (see figure below).
4. Select rule category and click ***Add new rule***. A dialog window is displayed (see figure below).

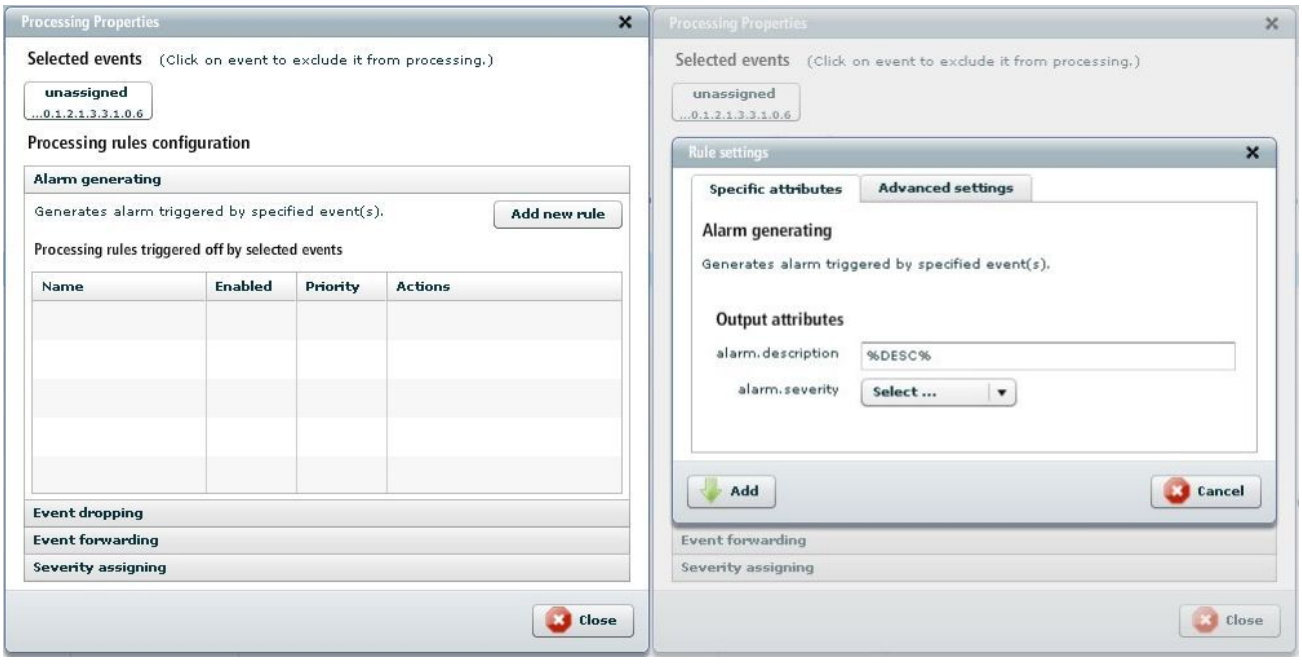


Figure 11: Creating custom processing rule

The newly created event processing rule is now visible and active (there's no need to logout).

5. MySQL NMS plugin overview

Features

Verax NMS provides MySQL application pluggable module to extend core built-in functionalities and allows MySQL database monitoring in various aspects like:

- Advanced views: General information, Status variables, System variables, Databases list.
- Availability monitoring
- Predefined performance counters templates
- User-defined counters and sensor based on SQL query

General information view

In the General tab, one may find the overall information about the version of the database, storage engine, version compile and uptime.

Server status view

Server status view provides information concerning server status – variable names and their values.

System status view

System status view provides information concerning system status – variable names and their values.

Databases view

The databases tab, presents detailed information regarding particular databases. After choosing the scheme name, one may see specific information about the database (table name, engine, rows, data length, index length, time of creation, update time) gathered in a form of a table.

Summary

If you performed all actions described in chapters 1-5 you are now able to monitor MySQL database.

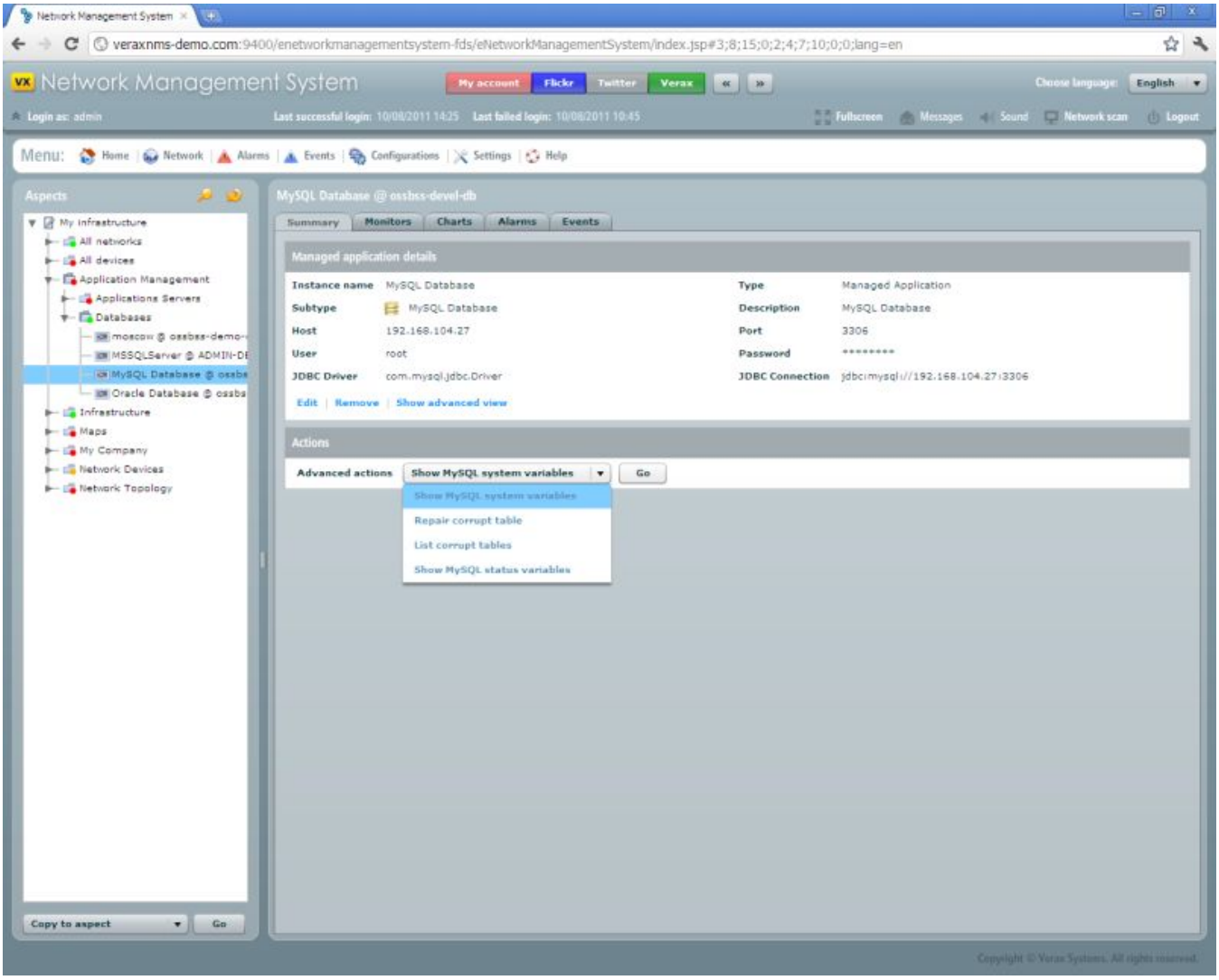


Figure 12: Showing MySQL system variables