Monitoring Oracle database with Verax NMS
Table of contents

Abstract ......................................................................................................................................... 3
1. Adding an Oracle database to device inventory ................................................................. 4
2. Adding sensors for Oracle database .................................................................................. 7
3. Adding performance counters for Oracle database ........................................................... 9
4. Creating custom event processing rules for Oracle database .......................................... 11
5. Oracle RDBMS plugin overview ....................................................................................... 12
Summary ..................................................................................................................................... 18
Abstract
This publication provides an overview on how to monitor and manage Oracle 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:
- **Oracle Database**: www.oracle.com/us/products/database
- **Monitoring tool**: www.veraxsystems.com/en/products/nms

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

In order to include Oracle 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 an Oracle 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 the Oracle database instance from the left-side *aspects* view.

![Aspect hierarchy tree](image)

**Figure 1: Aspect hierarchy tree**

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

![Actions](image)

**Figure 2: Adding a new managed application**

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

![Figure 4: Adding an Oracle database – parameters dialog](image)

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.
- **SID** - Oracle System Id.
- **User** - Username used to connect to the database.
- **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 Oracle database. Since, (in this example) sensors and performance will be added manually – click **No**.

The initial set of Oracle 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.
The newly added database is now visible in the aspect tree within the host’s node in Managed Applications category.

Figure 5: Oracle database properties window
### 2. Adding sensors for Oracle database

An Oracle 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 Oracle sensors (or JDBC sensors alternatively).
- Collecting performance data by configuring performance counters using either SQL-based Oracle counters (or JDBC counters alternatively) or SQL-based Oracle counter templates.

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 a number of pre-configured sensors. The following types of sensors for Oracle database application type are available by default:

- **Oracle** - checks if a given SQL query run on an Oracle database returns the expected result.
- **Oracle diagnostics** – detects abnormal database conditions.
- **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** (an Oracle 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 6: Adding a sensor](image)
4. Select the sensor you want to add and click **Next**.

![Figure 7: Adding sensor parameters](image)

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 Oracle 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 Oracle database performance counters, 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 **Create counter** from the global action menu and click **Go**. Select the counter you want to create and click **Next**.

![Figure 8: Adding counter for an Oracle database](image)
4. Once the data has been loaded, the edit window shows up with all counter attributes to be provided.

![Figure 9: Specifying counter parameters]

5. Specify the rest of counter parameters (alarms, thresholds etc.) and click **Finish**.

The new counter has been created and is now visible on the counter list.
4. Creating custom event processing rules for Oracle 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 we will show how to assign basic event processing rules:

- 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 10: Creating custom processing rule

5. The newly created event processing rule is now visible and active (there’s no need to logout).
5. Oracle RDBMS plugin overview

Verax NMS Oracle RDBMS plugin supports multiple instances (single or multiple hosts) of Oracle 9i and higher running in 32 and 64 bit environments.

General information view

- General database information including: open mode, instance state & status, creation and startup times.
- Dashboard view with charts for CPU usage and number of connecting processes.

![Figure 11: Oracle RDBMS plugin – general view](image)

Instance parameter view

Provides a comprehensive parameter list obtained from Oracle v$parameters, including: name, current value, description and number of revisions with changelog history.

**NOTE**

The changelog is only available if Oracle administrative console is configured for the database and SYSMAN account is present.
Tablespace view

The tablespace view provides a summary of all tablespaces in the database including name, status, file, extent and segment management information, size and usage statistics. Warning and critical usage levels are graphically indicated.
Datafile view

A comprehensive summary of data files, including:

- Datafile paths and names
- Tablespace associations
- Parameters: auto extensibility, maximum bytes allowed, number of blocks, maximum blocks allowed, status, online status
- Usage information: bytes allocated, bytes free with graphical visualization

For each datafile, the following performance statistics are available:

- Average read time
- Average write time
- Number of physical reads
- Number of physical writes
- Number of physical block reads
- Number of physical block writes

![Figure 14: Oracle RDBMS plugin – Datafiles view](image-url)
Monitoring Oracle database with Verax NMS

SGA/PGA memory

SGA:

- Memory management type and total size
- Detailed pool list including: pool name, size and usage

PGA:

- Total size, currently allocated, maximum allocation, hit ratio (%)
- Graphical visualization of PGA area effectiveness (optimal, one pass, multi pass) indicating effectiveness of SQL queries

![Oracle RDBMS plugin – memory view](image)

Performance view

Provides statistics for identifying database performance bottlenecks including:

- **Queries**: top SQL queries including id, query text, number of executions, total (summarized) execution time - essential for query tuning and application optimization
- **Sessions**: SID, terminal, user, program CPU usage, Disk I/O - allowing to identify top database users
- **Largest objects**: name, type, owner, tablespace, size in MB occupied by an object. Excessive objects are identified and visualized
Performance view

Provides active sessions summary including:

- User name
- Connecting host
- Program name
- State and blocking status
Diagnostics view

The view allows for automatic detection of abnormal database conditions examining the following values:

- Redo switch log
- TEMP space usage (%)
- Sort area size
- Number of commits per minute
- Rate at which rollbacks are being generated
- Number of invalid objects
- Number of disabled triggers
- Number of disabled constraints
- Number of snapshot errors
- Number of unanalyzed tables with detailed view
- Alive status of background processes

![Figure 18: Oracle RDBMS plugin – diagnostics view](image-url)
Summary
If you performed all actions described in chapters 1-5 you are now able to Oracle database.