



Configuring alerts in Denodo

Revision 20180329

NOTE

This document is confidential and proprietary of **Denodo Technologies**. No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2020
Denodo Technologies Proprietary and Confidential

CONTENTS

1 INTRODUCTION.....	4
2 JMX NOTIFICATIONS.....	5
2.1 WHEN TO USE.....	5
2.2 DESCRIPTION.....	5
2.3 STEPS TO CONFIGURE.....	5
3 LOG4J ALERTS.....	11
3.1 WHEN TO USE.....	11
3.2 DESCRIPTION.....	11
3.3 STEPS TO CONFIGURE.....	11
4 SNMP TRAPS.....	13
4.1 WHEN TO USE.....	13
4.2 DESCRIPTION.....	13
4.3 STEPS TO CONFIGURE.....	13
5 DENODO SCHEDULER HANDLERS.....	15
5.1 WHEN TO USE.....	15
5.2 DESCRIPTION.....	15

1 INTRODUCTION

This document explains how to send notifications by mail regarding events occurred in Denodo Virtual DataPort.

In some situations we need to be alerted if a specific event takes place in Denodo. In order to achieve that, we can use different techniques detailed in this article: log4j appenders, JMX alerts, SNMP traps and Denodo Scheduler Handlers.

2 JMX NOTIFICATIONS

2.1 WHEN TO USE

To register alerts regarding **Memory/CPU usage** information or any other information available in the **MBeans** exposed through JMX by the Denodo Platform.

2.2 DESCRIPTION

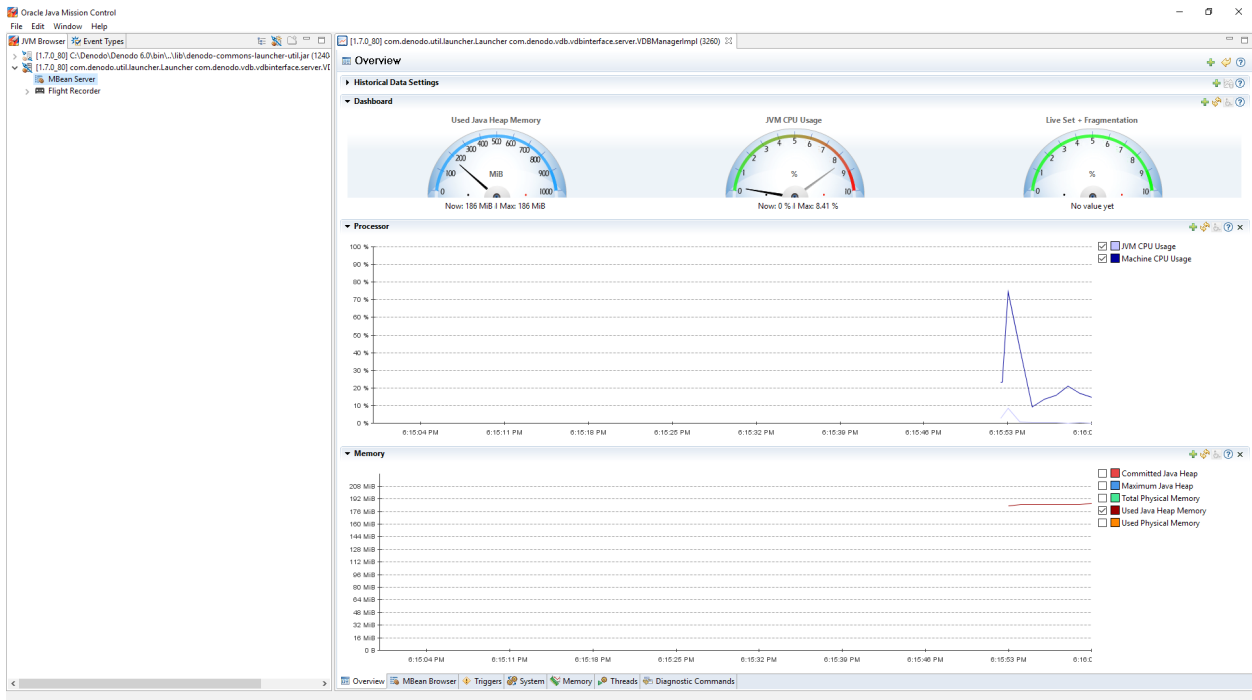
[Java Mission Control](#) (JMC) is an Oracle tool for managing, monitoring, profiling, and troubleshooting Java applications. We will see how to use this tool in order to monitor the Denodo Platform MBeans and to generate alerts based on their status.

2.3 STEPS TO CONFIGURE

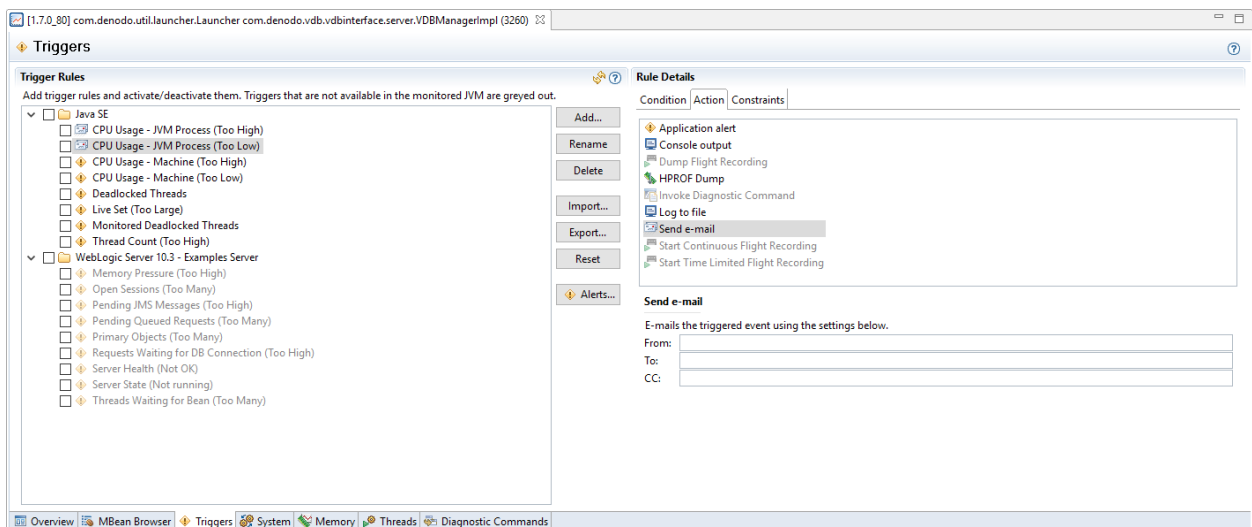
JMC is included as part of the Java distribution, if a JDK is already installed JMC can be executed (otherwise the Java Development Kit needs to be installed). Once the tool is running, perform the following steps:

1. Execute jmc.exe located at <JDK_PATH>/bin.
2. The "JVM Browser" shows the processes running locally. When the VDP Server is running in a remote machine you can connect with a remote JVM using the option "File" > "Connect" > "Create a new connection".
3. From the JVM Browser search for the VDP Server process. You can identify the process by the process ID (searching previously for this number) or identifying the element which command line starts as following:

```
com.denodo.util.launcher.Launcher  
com.denodo.vdb.vdbinterface.server.VDBManagerImpl...
```

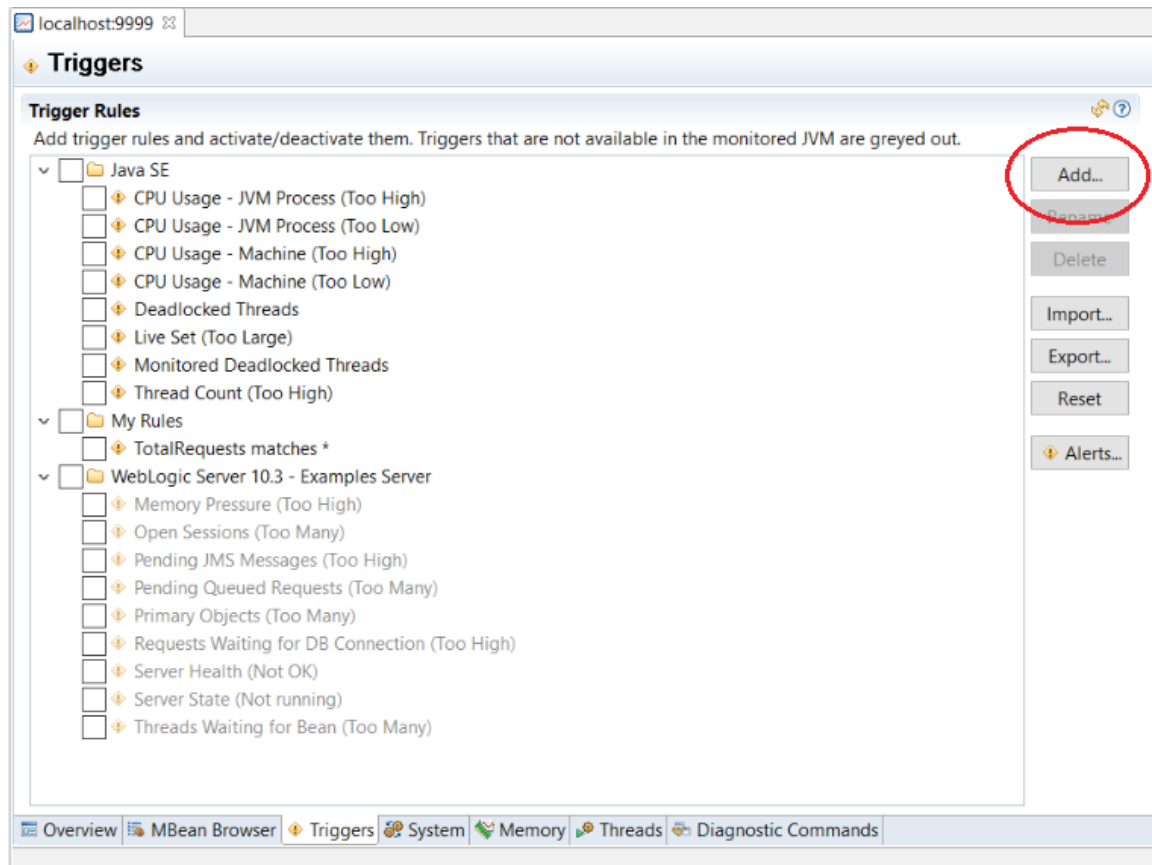


4. By clicking on the JVM Browser element and double clicking on “MBean Server” JMC will connect to the VDP Server.
5. On the MBeanServer wizard a **Triggers** tab is available at the bottom. This will open a wizard where trigger rules can be configured (activated and deactivated) for several events like CPU usage or Deadlocked threads. New trigger rules can also be created based on the exposed MBeans.

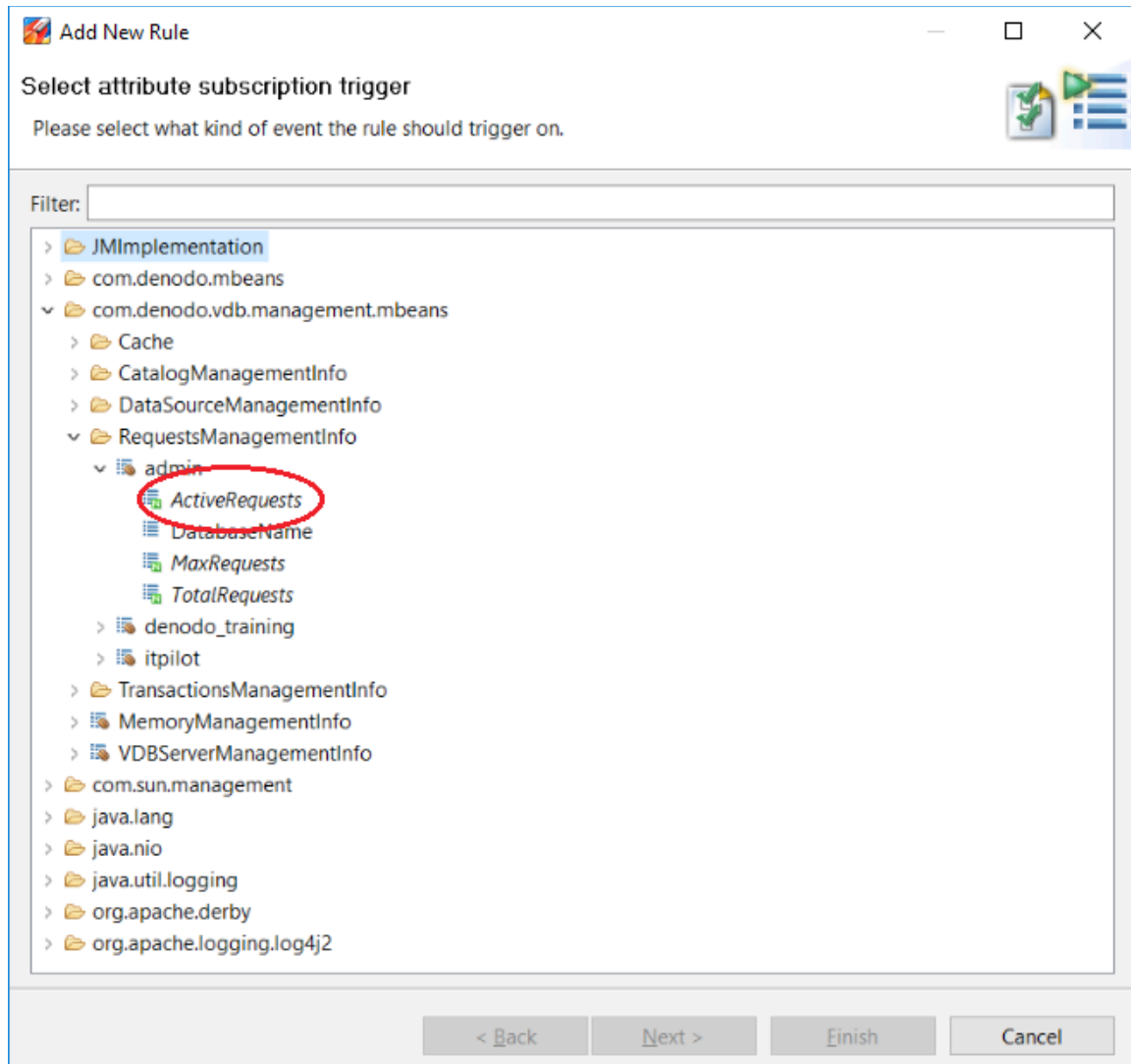


As an example we are going to create a new Trigger that sends an email every time that the number of active requests exceeds a limit number. The steps will be the following:

- a. On the Triggers tab click on “Add...” button.



- b. Search for the MBean containing the information and click on "Finish". In the KB article "[Monitoring Denodo servers with JMX](#)" more information about the available MBeans can be found.



- c. A new wizard called “**Rule Details**” will show up. In that wizard the values for triggering the condition can be setted. In this example the “current value” will be the limit that, when reached, the email will be sent.

Rule Details

Condition | **Action** | Constraints

Description

MBean Path:

Attribute Name:

Current Value:

Max trigger value:

Sustained period:

Limit period:

Trigger when condition is met.

Trigger when recovering from condition

- Once the condition is configured by clicking on the “Action” tab the “Send e-mail” action can be configured.

Rule Details

Condition | **Action** | Constraints

- Application alert
- Console output
- Dump Flight Recording
- HPROF Dump
- Invoke Diagnostic Command
- Log to file
- Send e-mail**
- Start Continuous Flight Recording
- Start Time Limited Flight Recording

Send e-mail

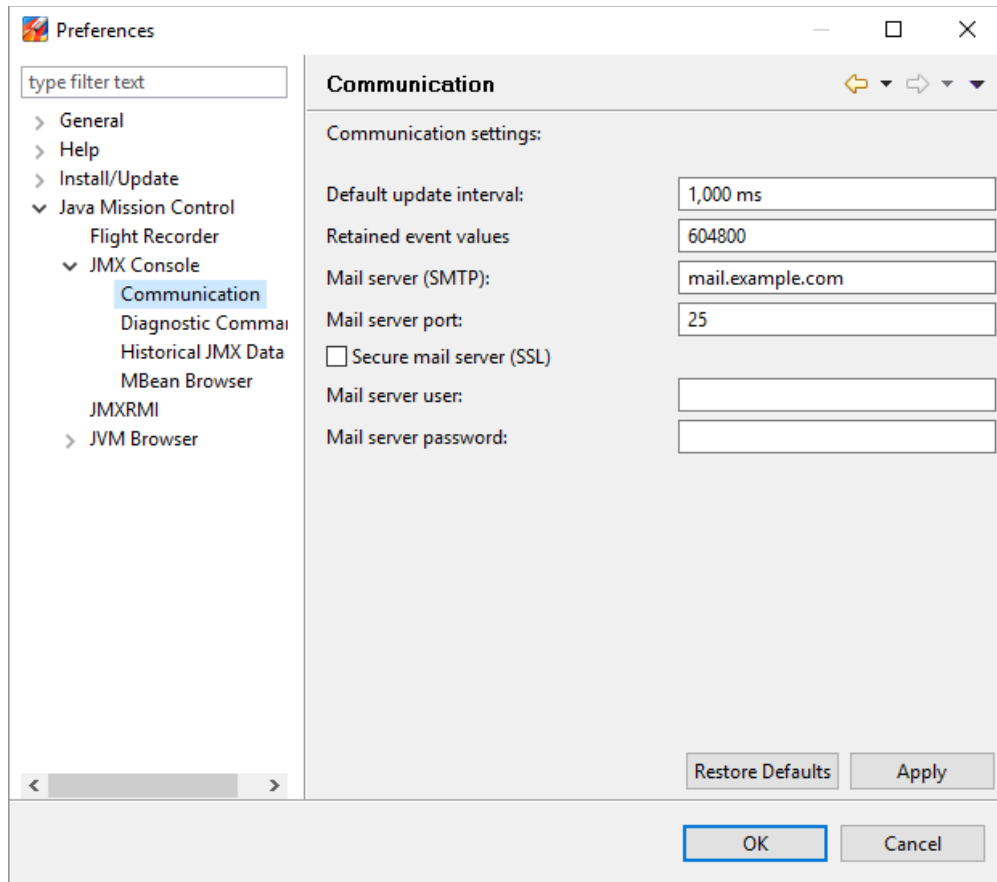
E-mails the triggered event using the settings below.

From:

To:

CC:

- In addition, the SMTP server has to be configured in order to be able to send the email. For that go to **Window > Preferences > Java Mission Control > JMX Console > Communication** and set up the SMTP Server configuration.



3 LOG4J ALERTS

3.1 WHEN TO USE

To get an e-mail alert of an event logged in the Denodo Virtual DataPort logs.

3.2 DESCRIPTION

Virtual DataPort uses the logging library **Apache Log4j** to log the activity of the Virtual DataPort server and its Administration Tool. Log4j allows defining the log level (TRACE, DEBUG, INFO, WARN, ERROR or FATAL) for different categories. The configuration of Log4j for the Virtual DataPort server is controlled by the file:

```
<DENODO_HOME>/conf/vdp/log4j2.xml
```

For the administration tool by the file:

```
<DENODO_HOME>/conf/vdp-admin/log4j.xml.
```

In these files you can find three main components: Appenders, PatternLayouts and Loggers.

A logger defines a log level for a category and they can be assigned to appenders previously configured. For instance:

```
<RollingFile name="VDBOUT" fileName="../../../logs/vdp/vdp-cache.log">
  <PatternLayout pattern="[%t] %d{yyyy-MM-dd'T'HH:mm:ss.SSS}"/>
</RollingFile>

<Logger name="com.denodo.vdb" level="error">
  <AppenderRef ref="VDBOUT" />
</Logger>
```

This configuration indicates that the category "com.denodo.vdb" on an error event will write the output of the log to the file configured in the VDBOUT appender following the defined pattern layout.

The **SMTPAppender** is an appender which sends an e-mail when a specific logging event occurs, typically on errors or fatal errors. Therefore, instead of writing the output to a file it is possible to configure a destination email address.

3.3 STEPS TO CONFIGURE

1. Configure the appender. The appender contains the information related to the SMTP server used for sending the email. Take into account that the configuration is slightly different for log4j2 and log4j:
 - a. For **log4j2** (VDP Server): <DENODO_HOME>/conf/vdp/log4j2.xml

```
<SMTP name="EMAILALERT"
  subject="VDP Log Alert"
  to="toUser@mydomain.com"
  from="fromUser@mydomain.com"
  smtpHost="mail.mydomain.com"
```

```

smtpPort="25"
smtpProtocol="smtp"
smtpUsername="mysmtusername"
smtpPassword="mysmtppassword"
smtpDebug="true"
bufferSize="512">
</SMTP>

```

- b. For **log4j** (Admin Tool): <DENODO_HOME>/conf/vdp-admin/log4j.xml
- ```

<appender
 name="EMAILALERT"
 class="org.apache.log4j.net.SMTPAppender">
 <param name="SMTPHost" value="mail.mydomain.com" />
 <param name="SMTPUsername" value="mysmtusername" />
 <param name="SMTPPassword" value="mysmtppassword" />
 <param name="From" value="fromUser@mydomain.com" />
 <param name="To" value="toUser@mydomain.com" />
 <param name="Subject" value="VDP Log Alert"/>
</appender>

```

2. Configure the loggers. Once the appender is configured, any logger with a log level for a category can be configured indicating the email appender as output:

```

<Logger name="com.denodo.vdp" level="FATAL">
 <AppenderRef ref="EMAILALERT" />
</Logger>

```

## 4 SNMP TRAPS

---

### 4.1 WHEN TO USE

To report the output of the Denodo Monitor.

### 4.2 DESCRIPTION

The Denodo Monitor Tool can be configured for sending Simple Network Management Protocol Traps (SNMP Traps) with the information received by the Queries Monitor or the Cache Monitor. Then using a SNMP Manager (or listener) you will be able to read the SNMP Trap Messages and configure alerts based on them.

The Denodo Monitor must be configured as an SNMP agent which sends the traps to the configured port. Then we have to run a SNMP manager or listener for reading the information sent by the Denodo Monitor and trigger messages based on events.

### 4.3 STEPS TO CONFIGURE

First of all, to configure the Denodo Monitor to send SNMP traps with the output of the Virtual DataPort query monitor and/or to send SNMP traps with the output of the Virtual DataPort cache monitor:

1. Open the file:  
`<DENODO_HOME>\tools\monitor\denodo-monitor\conf\ConfigurationParameters.properties`
2. To send SNMP traps with the output of the Virtual DataPort query monitor:
  - a. Set "vdpqueries.snmptrapagent.enable" to "true".
  - b. Set the values of the other "vdpqueries.snmptrapagent" properties to point to your SNMP listener.
3. To send SNMP traps with the output of the Virtual DataPort cache monitor:
  - a. Set "vdploadcacheprocesses.snmptrapagent.enable" to "true".
  - b. Set the values of the other "vdploadcacheprocesses.snmptrapagent" properties to point to your SNMP listener.

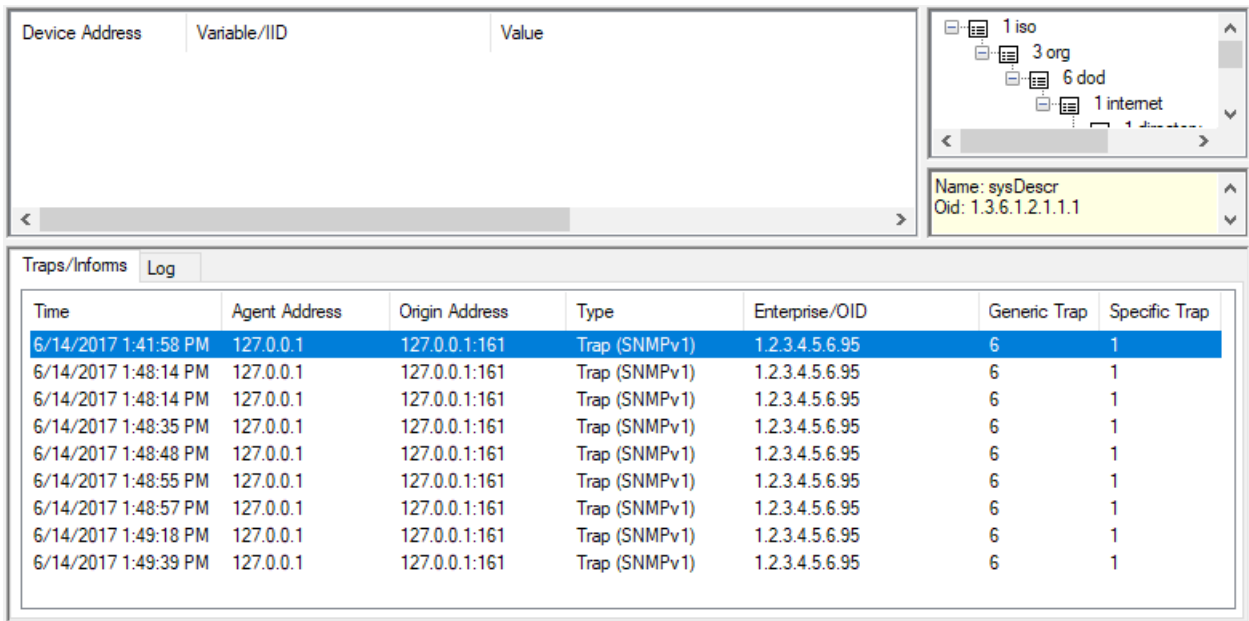
Either for step 2 and/or step 3 the properties are the same (uncomment to use the default values):

- Version: version of the SNMP protocol.
- Community: name of the group of computers that will receive SNMP traps.
- LocalIPAddress: IP address of the agent generating the SNMP trap.
- Targetaddress: IP address of the computer receiving the SNMP traps
- Targetport: port where the computer receiving the SNMP traps is listening.
- ApplicationOID: Application Object Identifier.
- EnterpriseOID: Enterprise Object Identifier

After enabling the "snmptrapagent" in the monitor configuration, a client (listener) to receive the SNMP Trap Alerts has to be configured.

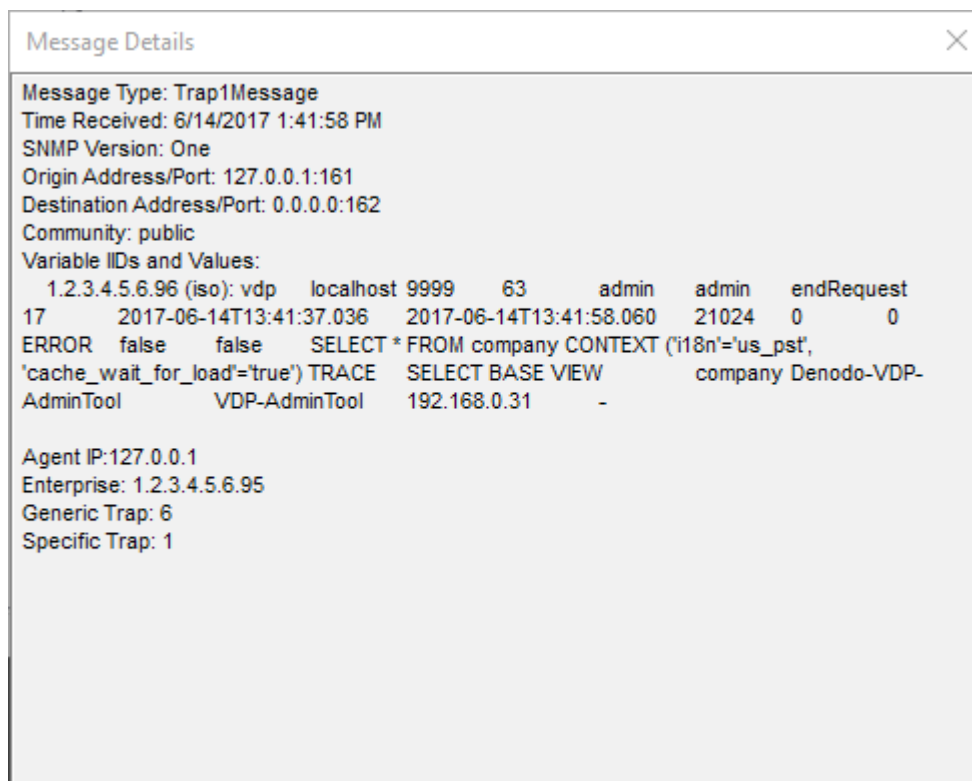
In the market there are several tools for monitoring SNMP Traps like Nagios or Zenoss with many functionalities using these tools that can trigger alerts based on trap events.

In this article a [simple SNMP tool](#) was used, however, the idea is basically the same for all of them: listen to the trap messages.



The screenshot shows a network management interface. At the top, there is a table with columns: Device Address, Variable/IID, and Value. Below this is a tree view showing a network hierarchy: 1 iso, 3 org, 6 dod, 1 internet, and 1 distro. A dropdown menu shows 'Name: sysDescr' and 'Oid: 1.3.6.1.2.1.1.1'. Below the tree view is a 'Traps/Infoms' section with a 'Log' tab. The log contains a table of trap messages:

| Time                 | Agent Address | Origin Address | Type          | Enterprise/OID | Generic Trap | Specific Trap |
|----------------------|---------------|----------------|---------------|----------------|--------------|---------------|
| 6/14/2017 1:41:58 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:48:14 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:48:14 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:48:35 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:48:48 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:48:55 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:48:57 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:49:18 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |
| 6/14/2017 1:49:39 PM | 127.0.0.1     | 127.0.0.1:161  | Trap (SNMPv1) | 1.2.3.4.5.6.95 | 6            | 1             |



The screenshot shows a 'Message Details' window with the following information:

Message Type: Trap1Message  
 Time Received: 6/14/2017 1:41:58 PM  
 SNMP Version: One  
 Origin Address/Port: 127.0.0.1:161  
 Destination Address/Port: 0.0.0.0:162  
 Community: public  
 Variable IIDs and Values:  
 1.2.3.4.5.6.96 (iso): vdp localhost 9999 63 admin admin endRequest  
 17 2017-06-14T13:41:37.036 2017-06-14T13:41:58.060 21024 0 0  
 ERROR false false SELECT \* FROM company CONTEXT ('i18n'='us\_pst',  
 'cache\_wait\_for\_load'='true') TRACE SELECT BASE VIEW company Denodo-VDP-  
 AdminTool VDP-AdminTool 192.168.0.31 -

Agent IP: 127.0.0.1  
 Enterprise: 1.2.3.4.5.6.95  
 Generic Trap: 6  
 Specific Trap: 1

## 5 DENODO SCHEDULER HANDLERS

---

### 5.1 WHEN TO USE

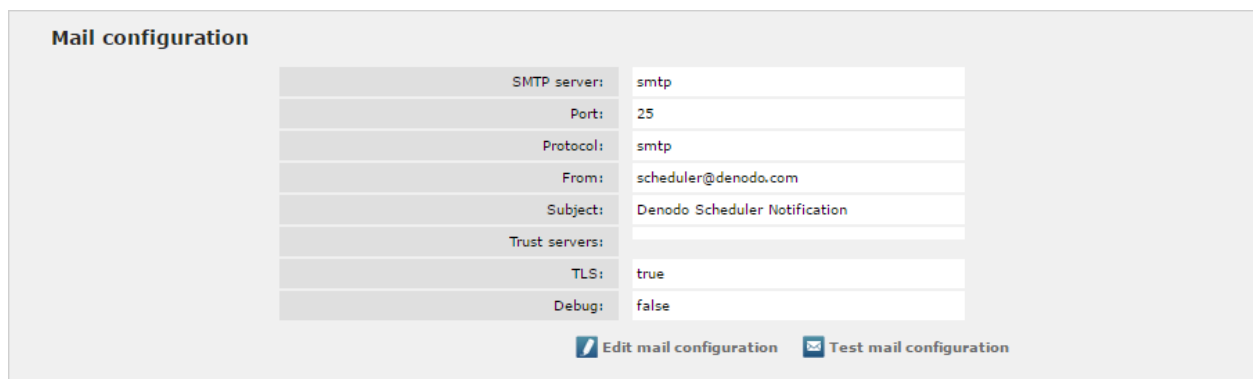
To send an email alert about the failure of a particular Job in Denodo Scheduler.

### 5.2 DESCRIPTION



When configuring Jobs in Scheduler it is possible to configure handlers. A handler is an action that is executed as the last step before completing the execution of a job (if the conditions are met). Denodo Scheduler includes as a built-in handler the “Mail” handler. This handler allows to send an email with the result report on the job execution. It requires a list of destination e-mail addresses to be specified and allows several conditions to be set (determining in which cases the mail has to be sent).

## Steps to configure

In order to be able to send the email Scheduler requires a SMTP server to be configured. This configuration has to be set in “Configuration” > “Server Configuration” > “Mail configuration”.

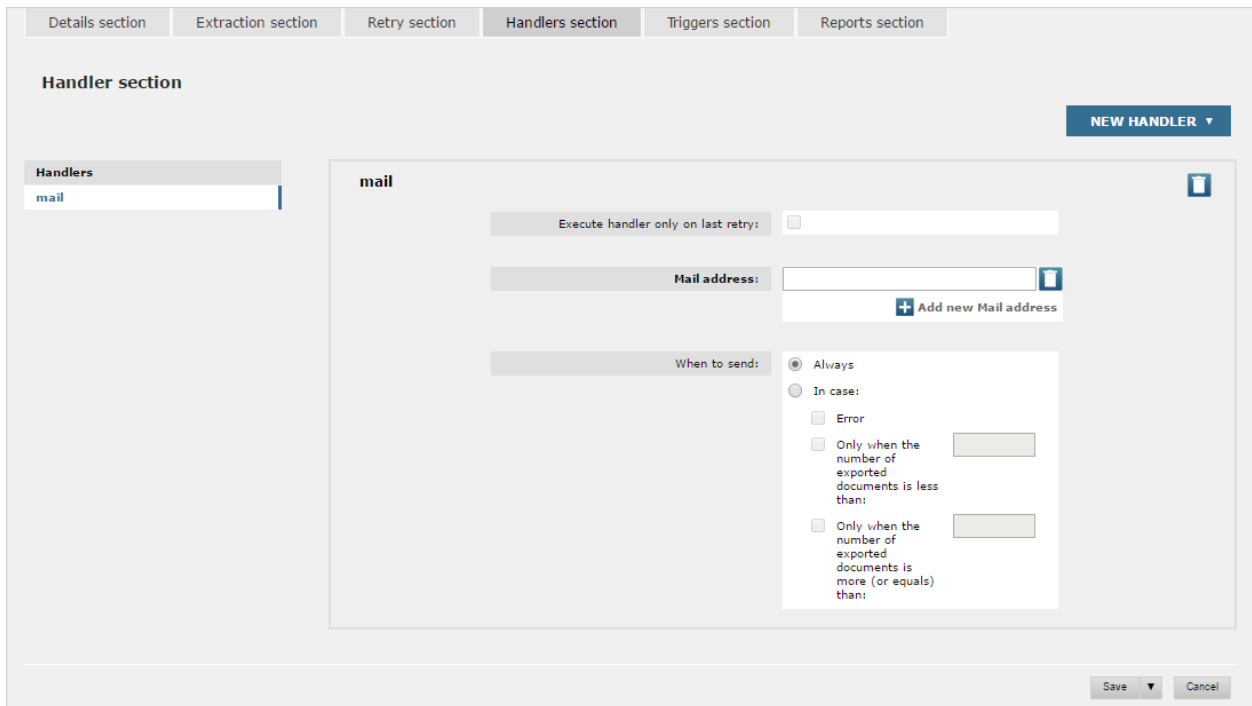


| Mail configuration |                               |
|--------------------|-------------------------------|
| SMTP server:       | smtp                          |
| Port:              | 25                            |
| Protocol:          | smtp                          |
| From:              | scheduler@denodo.com          |
| Subject:           | Denodo Scheduler Notification |
| Trust servers:     |                               |
| TLS:               | true                          |
| Debug:             | false                         |

 Edit mail configuration  Test mail configuration

Once the mail server is set you have to follow this steps:

1. Select the Job (or create a new one).
2. Click on the “[Handlers section](#)” of the Job configuration.
3. Select “New Handler” > “mail”.
4. Add the target mail addresses and the configuration “When to send”.
5. Save the Job.



The screenshot shows the 'Handler section' of the Denodo configuration interface. At the top, there are tabs for 'Details section', 'Extraction section', 'Retry section', 'Handlers section', 'Triggers section', and 'Reports section'. The 'Handlers section' is active, and a 'NEW HANDLER' button is visible in the top right. On the left, a 'Handlers' list contains 'mail'. The main area is titled 'mail' and contains the following configuration options:

- Execute handler only on last retry:**
- Mail address:**
- When to send:**
  - Always
  - In case:
    - Error
    - Only when the number of exported documents is less than:
    - Only when the number of exported documents is more (or equals) than:

At the bottom right, there are 'Save' and 'Cancel' buttons.

Next time the Job is executed, before finishing the execution, Scheduler will check if the “When to send” conditions are met and, if it is the case, it will send the Job report to the configured target emails.