Installation Instruction

VDS-II

SIP-active

CONET

Version 2012

Table of Content

1 Document History

Version	Date	Name	Changes
1.0	2016-11-28	AF	Initial draft
2012	2020-12-17	PE	VC Adaptation

2 General

2.1 Purpose

This document describes the installation VDS-SIP-active-CONET. It covers the installation process, the standard configuration and troubleshooting.

2.2 Acronyms and Abbreviations

API	Application Programming Interface
CRD	Call Related Data
DLL	Dynamic Link Library (Microsoft)
GUI	Graphical User Interface
RTP	Real-time Transport Protocol (RFC3550)
RTSP	Real Time Streaming Protocol (RFC2326)
TETRA	Terrestrial Trunked Radio
VDS-II	VoIP Decoding System (ATIS)
VEP	VoIP Export Protocol (ATIS)
REGEXP	Regular Expression

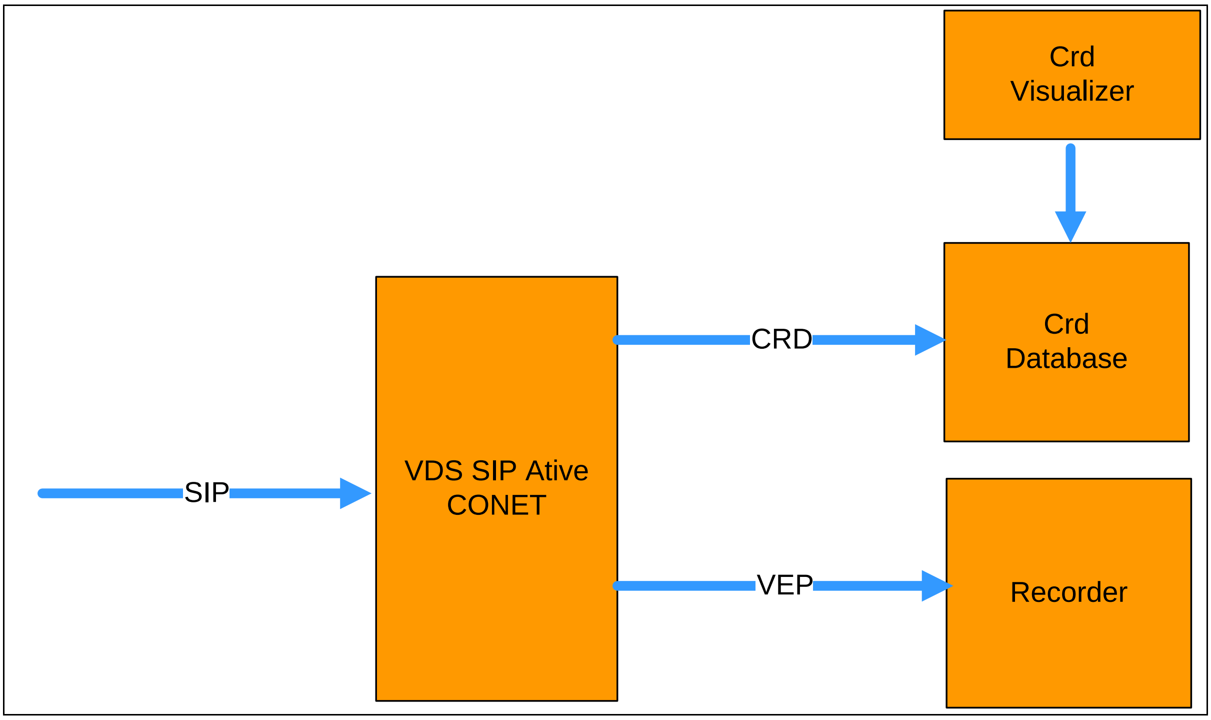
3 Introduction

The VDS-SIP-active CONET received call related data (crd) in the OPTIONS SIP message; this data is stored in an external crd Database. The Database used is mongoDB for his ability to store objects

4 Prerequisite

➔ MongoDB 3.x.x installed

5 VDS-II-SIP-Active-CONET



The VDS-II:

- handles SIP messaging
- extracts CONET Call related data and insert them into crd database

CrdVisualizer:

- Graphical tool to display call related data

6 Installation

6.1 Installation of VDS-II

The VDS-II package is provided as a Windows installer named **vds-sip-active-CONET-x.x.x _setup.exe** to be installed on ATIS Interface Computer. The screenshots below explain the most important steps.

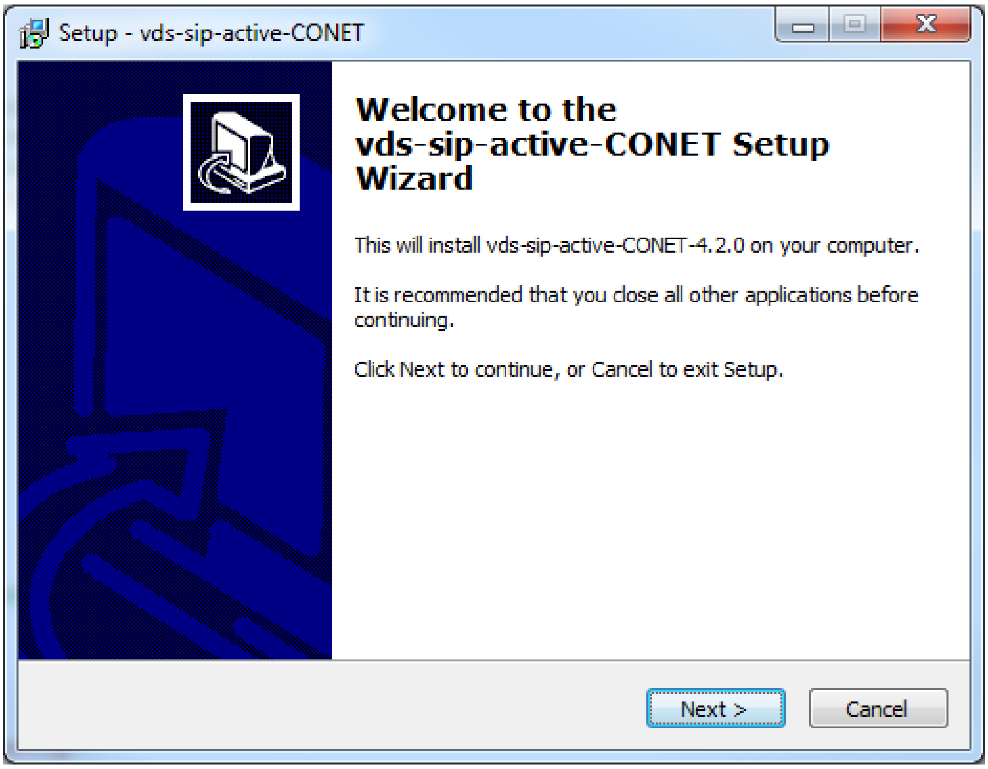
NOTE: A 32bit version of the Java Runtime Environment starting at java 1.6 must already be installed.

Typing in the command dos windows java –version can only give the current version not the list of installed versions.

The installer will check the registry to determine if a suitable version is present.

Locate the file vds- sip-active-CONET-x.x.x_setup.exe and double-click on the icon:

The following dialog box appears, press Next:



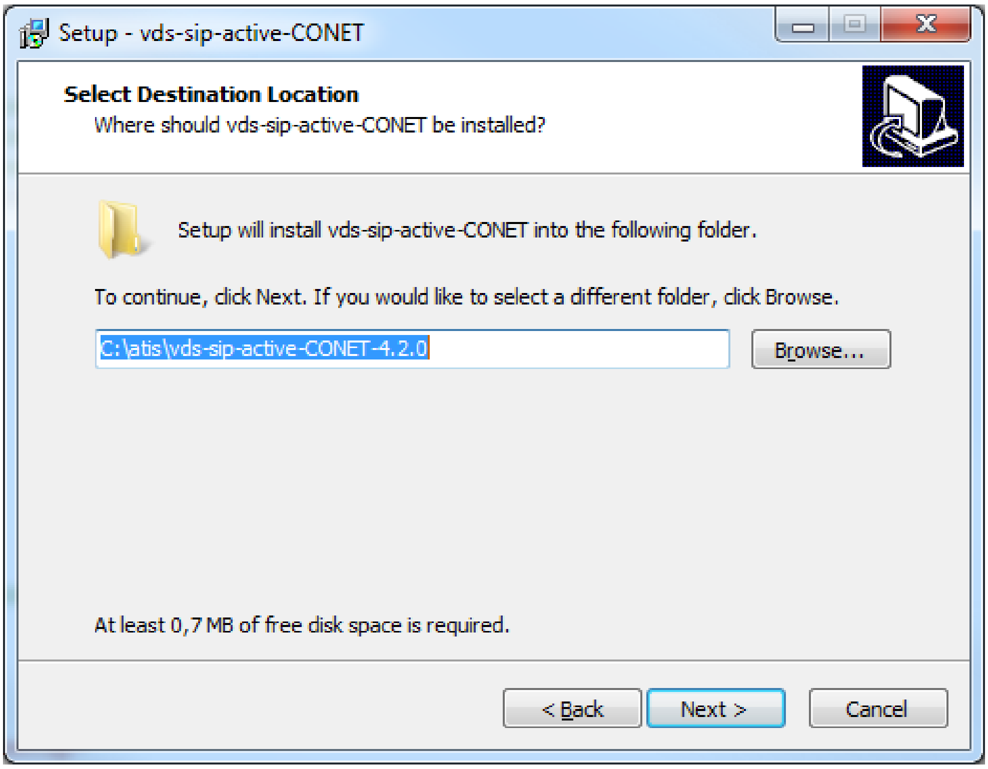
Welcome to the vds-sip-active-CONET Setup Wizard

This will install vds-sip-active-CONET-4.2.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Select the directory where VDS-II will be installed in:



Select Destination Location

Where should vds-sip-active-CONET be installed?

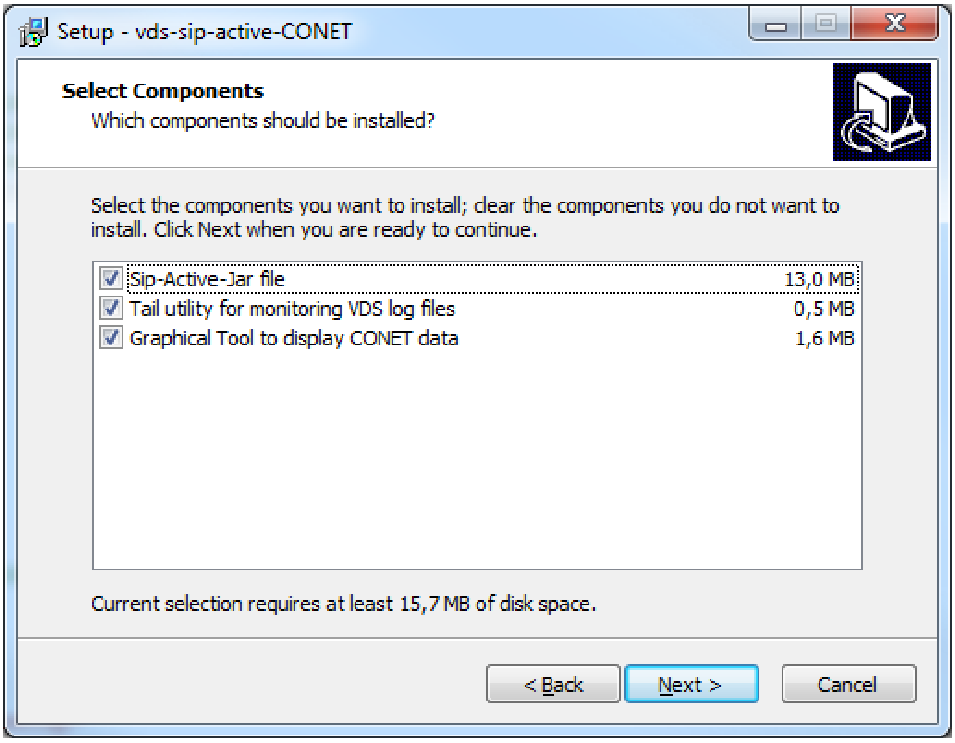
Setup will install vds-sip-active-CONET into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\atis\vds-sip-active-CONET-4.2.0

At least 0,7 MB of free disk space is required.

Select the components that will be installed:



Setup - vds-sip-active-CONET

Select Components

Which components should be installed?

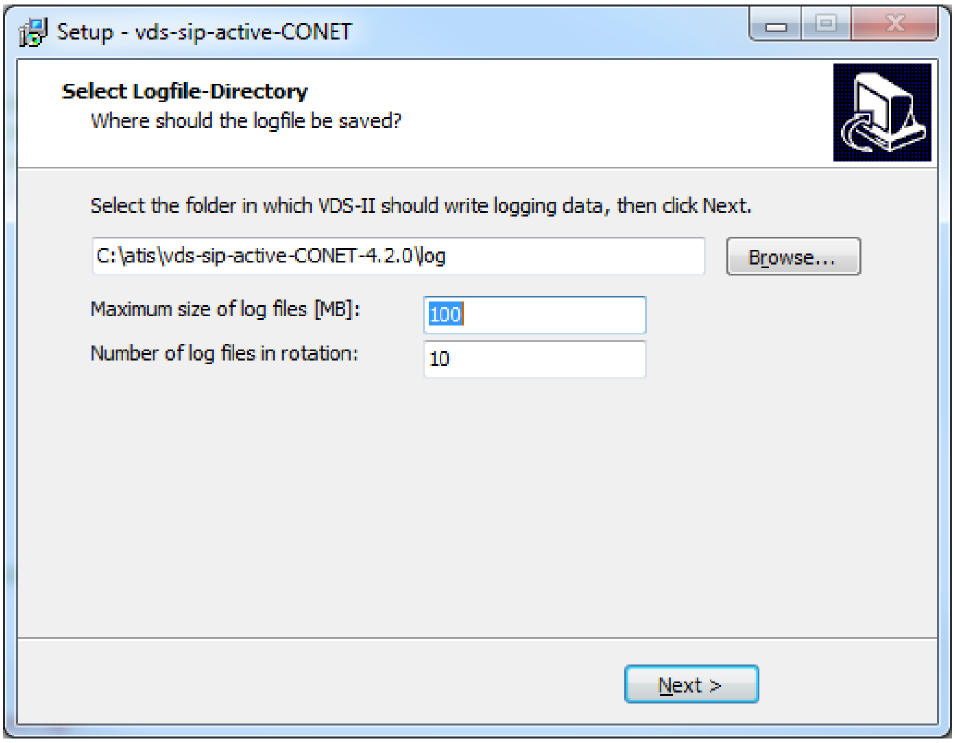
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☑ Sip-Active-Jar file	13,0 MB
☑ Tail utility for monitoring VDS log files	0,5 MB
☑ Graphical Tool to display CONET data	1,6 MB

Current selection requires at least 15,7 MB of disk space.

Sip-active-jar file is the main VDS-II software

Select the location for the log files, as well as the size and the number of rotated log files:



Setup - vds-sip-active-CONET

Select Logfile-Directory

Where should the logfile be saved?

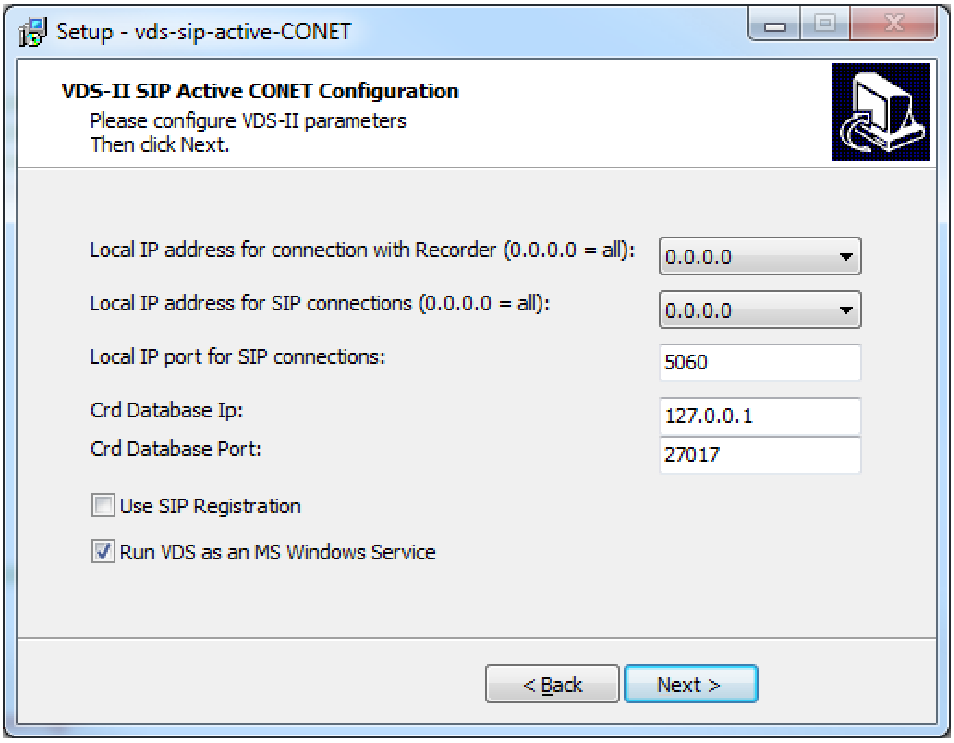
Select the folder in which VDS-II should write logging data, then click Next.

C:\atis\vds-sip-active-CONET-4.2.0\log

Maximum size of log files [MB]: 100

Number of log files in rotation: 10

6.2 VDS-II –Openscape-csta configuration parameter



VDS-II SIP Active CONET Configuration

Please configure VDS-II parameters
Then click Next.

Local IP address for connection with Recorder (0.0.0.0 = all): 0.0.0.0

Local IP address for SIP connections (0.0.0.0 = all): 0.0.0.0

Local IP port for SIP connections: 5060

Crd Database Ip: 127.0.0.1

Crd Database Port: 27017

□ Use SIP Registration

☑ Run VDS as an MS Windows Service

Local IP address for connection with recorder comboBox

- This Combobox is used to choose the IP of the VDS-II server where the recorder will be connected (this is the Ip which will be used in the Recorder configuration)

Local IP address for SIP connection

- This Combobox is used to choose the IP address of the VDS-II SIP server

Local IP port for SIP connections

- Default is 5060 (this is the default standard SIP port only change if you sure what you doing)

Crd Database IP

- Default is 127.0.0.1. The mongoDB database Ip

Crd Database Port

- Default is 27017 (only change if you sure what you doing)

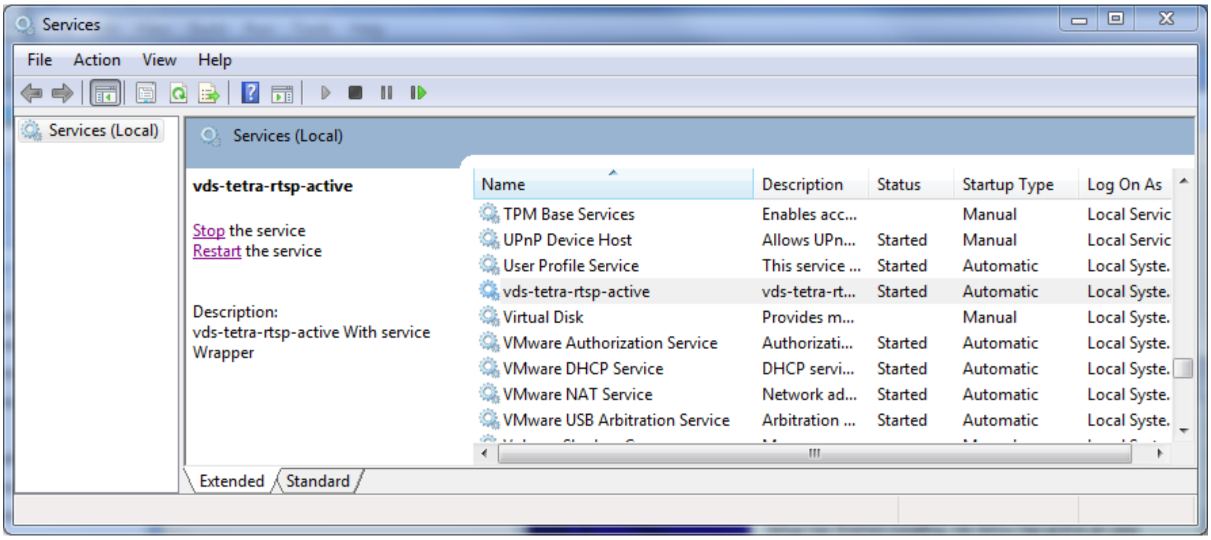
Use SIP registration

- This checkBox enables you to configure SIP registration

Run VDS-II as a service

- By default we recommend to run VDS-II as a service. If it is not checked the VDS-II will be run in console mode (be cautious any click in the console will freeze the VDS-II).

With MS Windows Service Manager, you can verify that the VDS-II service has been started:



vds-tetra-rtsp-active

Stop the service
Restart the service

Description:
vds-tetra-rtsp-active With service Wrapper

Name	Description	Status	Startup Type	Log On As
TPM Base Services	Enables acc...		Manual	Local Servic
UPnP Device Host	Allows UPn...	Started	Manual	Local Servic
User Profile Service	This service ...	Started	Automatic	Local Syste.
vds-tetra-rtsp-active	vds-tetra-rt...	Started	Automatic	Local Syste.
Virtual Disk	Provides m...		Manual	Local Syste.
VMware Authorization Service	Authorizati...	Started	Automatic	Local Syste.
VMware DHCP Service	DHCP servi...	Started	Automatic	Local Syste.
VMware NAT Service	Network ad...	Started	Automatic	Local Syste.
VMware USB Arbitration Service	Arbitration ...	Started	Automatic	Local Syste.

In case of problems, you can also check the log files in the log directory of the program.

6.3 CrdVisualizer

The CrdVisualizer tool is installed with the VDS-sip-active-CONET.

It is located where the VDS-II was installed double click on CrdVisualizer.jar to start it.

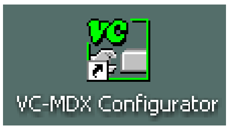
If you want to start it on another computer copy the CrdVisualizer.jar and the libs directory.

Java 1.8 need to be installed to be able to run it.

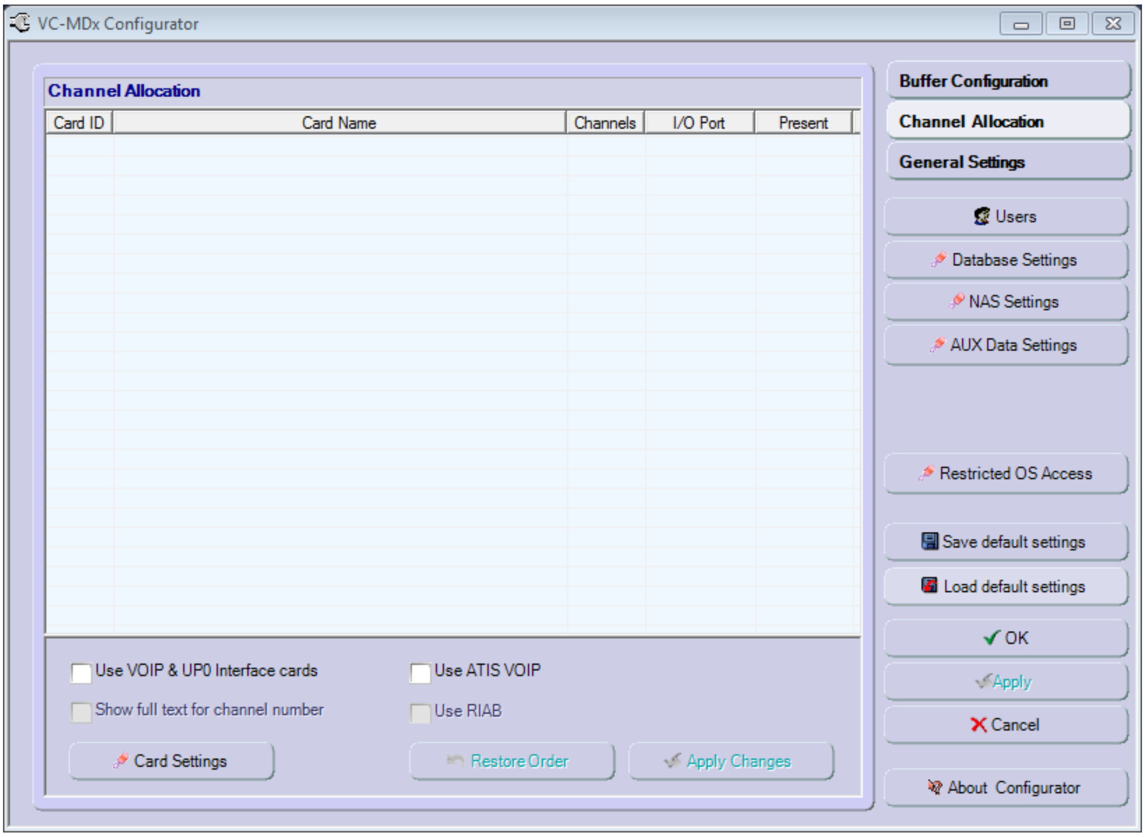
6.4 Configuration of VC-MDx Recorder

The MDX recorder must be configured before it can be used with the VDS-II application. This is done on the Recorder computer with the program VC-MDX Configurator.

NOTE: For configuring the Recorder, the VDS-II application must be running on the interface computer.



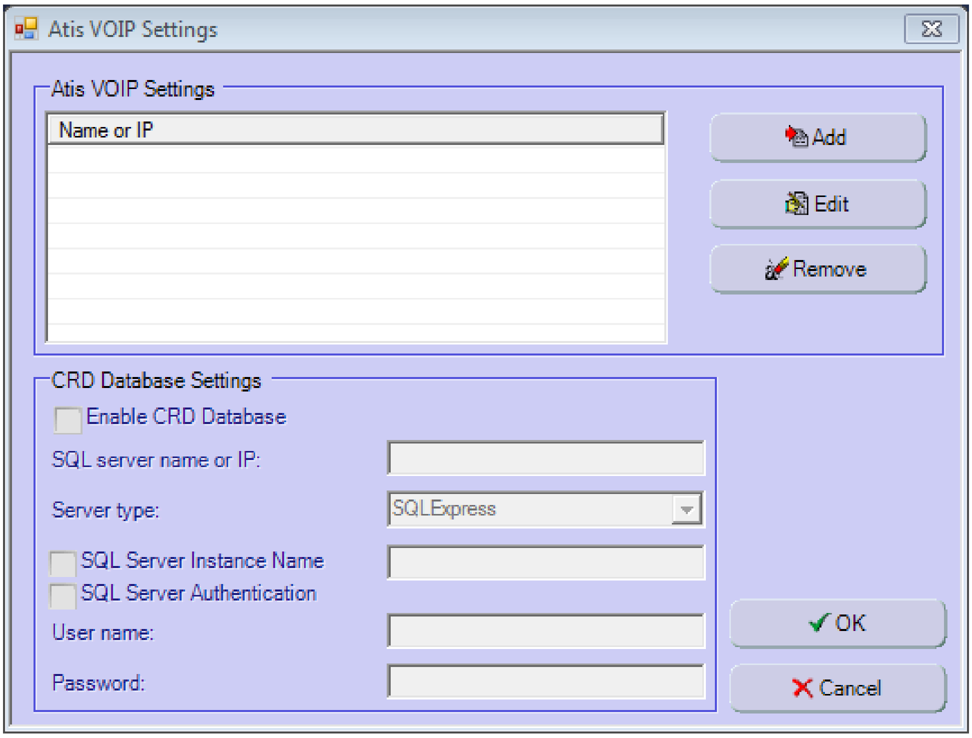
The following dialog box appears. Check the Use ATIS VOIP,a new dialog will pop up:



Card ID	Card Name	Channels	I/O Port	Present

□ Use VOIP & UP0 Interface cards
□ Use ATIS VOIP
□ Show full text for channel number
□ Use RIAB

The new dialog displays a list of connections to VDS-II applications (currently empty). Press the Add button:



□ Enable CRD Database

SQL server name or IP:

Server type: SQLExpress

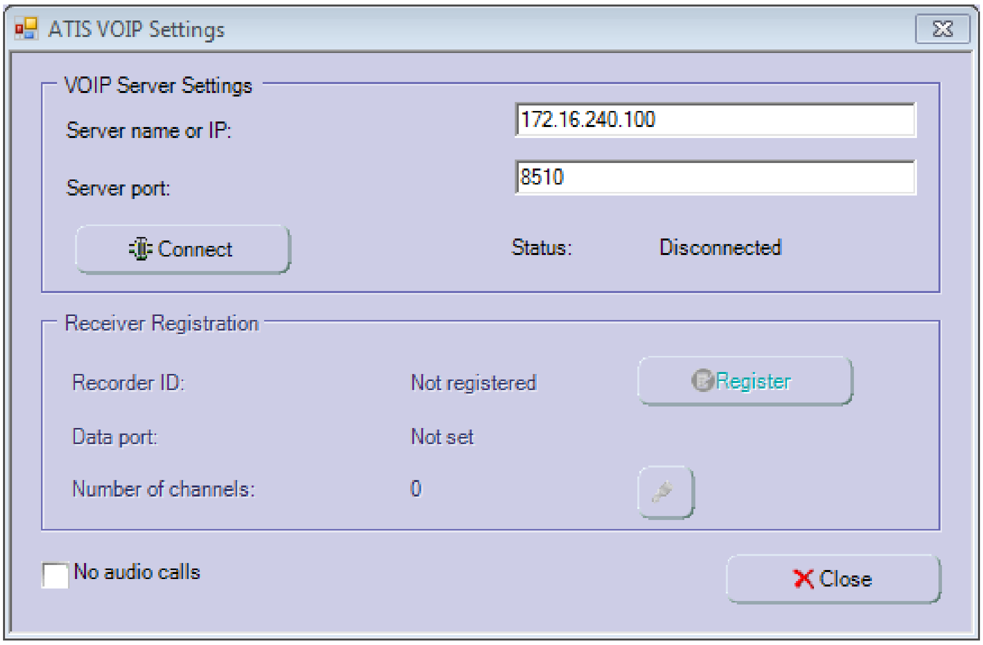
□ SQL Server Instance Name

□ SQL Server Authentication

User name:

Password:

Next dialog displays parameters for the VEP connection to VDS-II. In the Server name or IP field, enter the IP address of the Interface computer where VDS-II is installed. Then press the Connect button:



Server name or IP: 172.16.240.100

Server port: 8510

Status: Disconnected

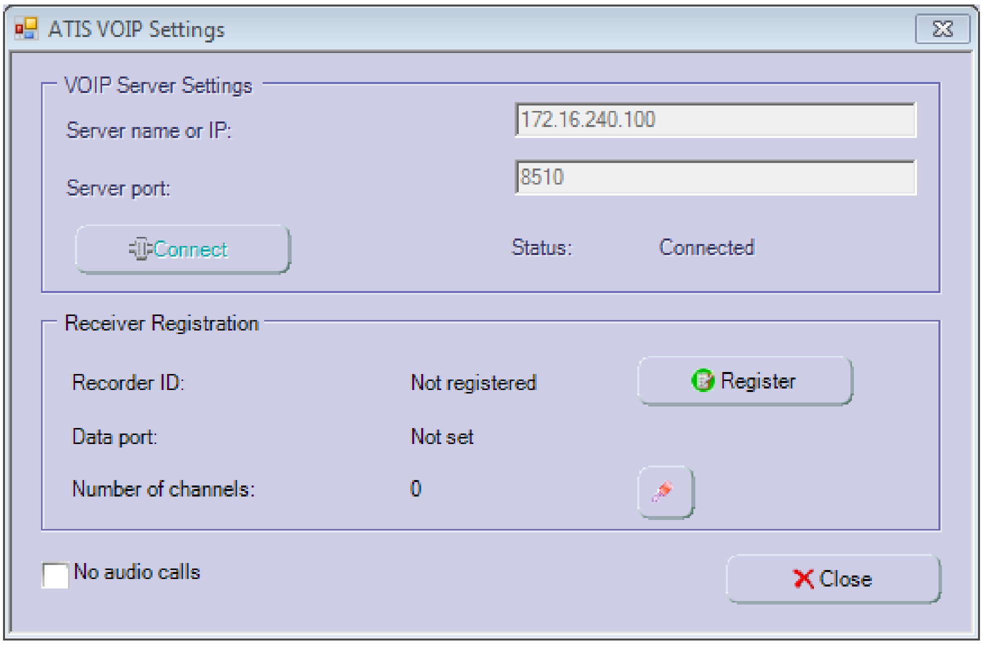
Recorder ID: Not registered

Data port: Not set

Number of channels: 0

□ No audio calls

Once the connection is established, press the Register button:



ATIS VOIP Settings

VOIP Server Settings

Server name or IP: 172.16.240.100

Server port: 8510

Status: Connected

Receiver Registration

Recorder ID: Not registered

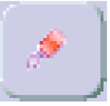
Data port: Not set

Number of channels: 0

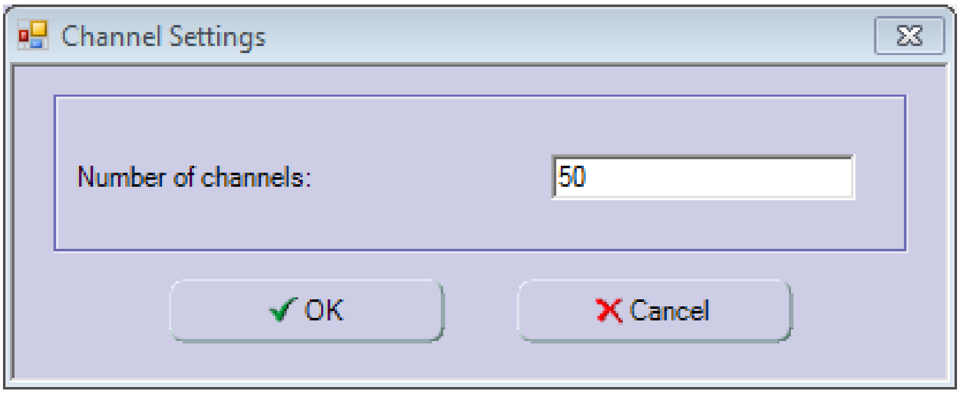
□ No audio calls

Note: The No audio calls checkbox allows for keeping records of calls without audio, for example rejected or discarded calls.

Finally, select the number of channels that should be used by the Recorder by pressing the button:



Following dialog appears where you can enter the number of channels:

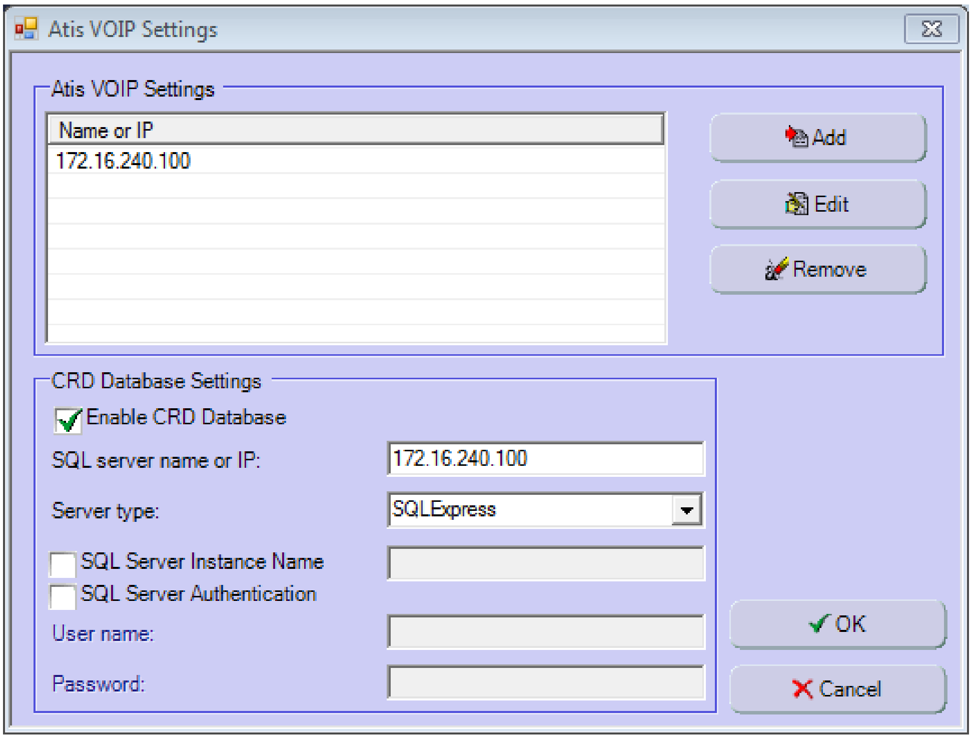


Channel Settings

Number of channels: 50

NOTE: The number of channels is object of ATIS licensing. Please, contact ATIS Sales for details.

The new VDS-II connection is now added and it is displayed in the list:



Name or IP: 172.16.240.100

☑ Enable CRD Database

SQL server name or IP: 172.16.240.100

Server type: SQLExpress

□ SQL Server Instance Name

□ SQL Server Authentication

User name:

Password:

Only VDS-II which are collecting enhanced/additional CRD (call related data) are using the CRD database

In order to activate the optional database that contains call events and related data, tick the checkbox Enable CRD Database

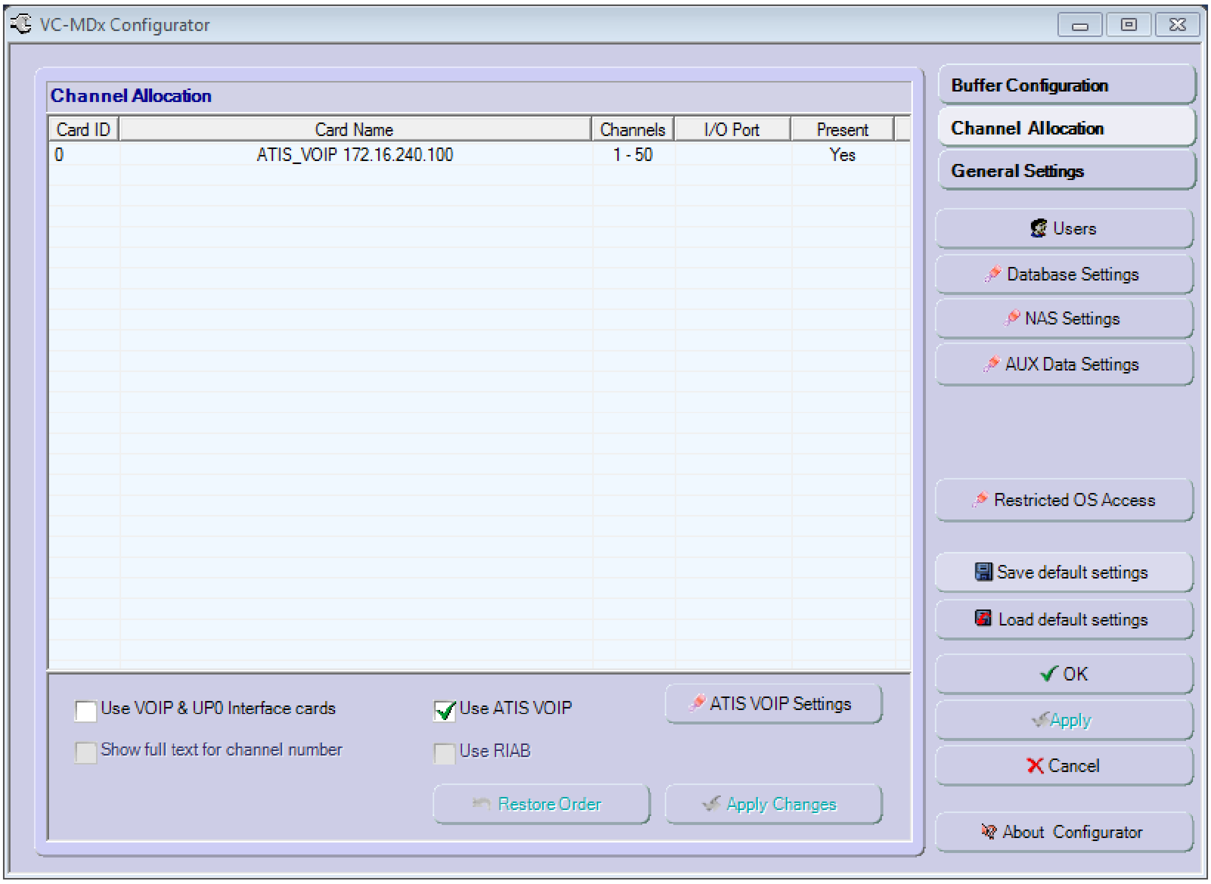
NOTE: For now there is no connection possible between the recorder and the CONET crd database.

Usually, this database is in the Recorder computer along with main record database, therefore the IP address of the Recorder can be entered in the field named SQL server name or IP.

If the CRD database runs in another instance of SQL Server, or if the authentication parameters are not the default ones, please fill related fields in.

Once everything has been configured, press the OK button.

There is a new ATIS_VOIP entry with the numbered range of channels in the list for channel allocation:



Channel Allocation

Card ID	Card Name	Channels	I/O Port	Present
0	ATIS_VOIP 172.16.240.100	1 - 50		Yes

☐ Use VOIP & UP0 Interface cards

☑ Use ATIS VOIP

☐ Show full text for channel number

☐ Use RIAB

You can use the ATIS VOIP Settings button to check details or change the parameters.

7 Configuration

7.1 Routing

The routing allows assigning specific calls (determined by filters) to specific recorder channel.

The drawback is that the assigned channel can only be used for calls which fulfill the filter criteria.

The routing configuration is done by editing and configuring the routing-configuration.xml file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<routingTable>
    <location>Any</location>
    <recorderList>
        <recorder name="R0001">
            <channelList>
                <channel id="1" filter="usera"/>
                <channel id="2" filter="*"/>
                <channel id="3" filter="*"/>
                <channel id="4" filter="sip:tel4500"/>
                <channel id="5" filter="*"/>
                <channel id="6" filter="*"/>
                <channel id="7" filter="172.16.240.91"/>
                <channel id="8" filter="172.16.240.91"/>
                <channel id="9" filter="172.16.240.91"/>
                <channel id="10" filter="/"/>
                <channel id="11" filter="/"/>
                <channel id="12" filter="/"/>
            </channelList>
        </recorder>
    </recorderList>
</routingTable>
```

Every channel is assigned a filter.

"*" means any value (that channel is taken as a dynamic one)

"/" means blocked channel.

Any other value in the filter is taken as a regular expression (regexp) and can provide extremely powerful matching but requires extreme care as the results could be unexpected.

The routing can be modified without the need of restarting VDS-II, the new routing will be taken into account on next successful call.

Multiples criterion can be assigned to the same channel.

By default the routing use the rule "contains" for the filters on UserId then CallingParty and last on CalledParty (the order and the fields can be configured in the spring-config.xml). Please refer to the document VDS-***II_Advanced-configuration_1.4.pdf***

7.2 Formatting

The formatting is done just before sending all data to the recorder. It allows the user to modify/cleanup/replace some fields.

An ip address can be replaced by a name:

"172.16.28.95" replaced by "Equipment Room"

A complex number can be reduced to a more readable format

"Sip:2356@172.16.28.95:4893" replaced by 2356

A comment can be translated

"Call held" replaced by "llamada en espera"

Please refer to the document VDS-II_Advanced-configuration_1.4.pdf

8 Installed Files

8.1 Spring-config.xml

This file describes the VDS-II software, some parameters can be modified manually but we recommend extreme care when editing that file.Normally everything is configured during installation and that file doesn't need to be edited / modified.

Some specific value can be adjusted in the spring-config.xml. It is recommended to use or modify these values with the support of ATIS.

8.2 Log4j.properties

This file contains the logger properties. It can be used to modify the log level of the console and the log level of the log file. Changes in that file can be done without restarting the VDS-II (it can take up to 30 seconds for the changes to become active).

8.3 Runtime.xml

This file contains the channels registered by the recorder. It should not be modified

8.4 Batch files

If you choose to start VDS-II as a service, two batch files are present

- install-VDS-II-eurocae-rtsp-active-service.bat Install the service
- remove-VDS-II-eurocae-rtsp-active-service.bat remove the service*

These files are just there for convenience if you want to start/stop VDS-II without the need to go through the Microsoft service panel.

8.5 Dll Files

These dll contains codecs used for transcoding/decoding audio streams.

8.6 Wrapper files

The wrapper is the software which allows the VDS-II to run as a service.

8.7 Jar file

This is the main VDS-II software

8.8 Log files

Two log files are created into the directory chosen during installation

vds-sip-active-CONET.log	VDS log file
wrapper.log	VDS console output and java runtime console

9 Troubleshooting

As the VDS-II is a service or running in a console it’s not easy to find out what’s going on when no recording or errors occur. The logs should contain hints about what happened.

You should look in the log file for specific errors.

9.1 Exceptions

An Exception is a problem that occurs during software execution.

The problem can be of several types, including:

- User entered invalid data (wrong configuration)
- Programming error (bug)
- Resource unavailable (network disconnected, resource busy etc..)

The Java runtime environment possesses exceptions handling processes.

The programmer can define his own Exceptions and the processing which is associated.

When an exception is caught by the software the execution of the code is rerouted to a specific error handling part.

Several Exceptions are caught and handled by the software as they are controlled during compilation (user defined Exceptions, error which cannot be foreseen like unexisting file etc…) these are called checked exceptions.

Runtime Exceptions are caught by the java runtime environment and most of the time could have been avoided by the programmer (bug).

The last types are errors and cannot be caught so they cannot be handled by the java software and happen on severe failures like OutOfMemoryError.

If some exceptions are not caught by the VDS-II software, but by the Java Runtime Environment they are appearing in the console.

If you are running VDS-II as a service they are logged into the file wrapper.log.

The printout should looks like:

java.lang.NullPointerException

- *at java.util.concurrent.ConcurrentLinkedQueue.offer(ConcurrentLinkedQueue.java:273)*
- *at java.util.concurrent.ConcurrentLinkedQueue.add(ConcurrentLinkedQueue.java:237)*
- *at com.atissystems.recorder.vds.core.RecordingAdapter.ioEvent(RecordingAdapter.java:344)*
- *at com.atissystems.vds.recording.RecordingLogic.ioEvent(RecordingLogic.java:80)*
- *at java.lang.Thread.run(Unknown Source)*

We cannot give all the possible exceptions (more than a hundred of them).

Here are the most common exceptions which should be reported to Atis as soon as possible

java.lang.NullPointerException

java.lang.OutOfMemoryError

java.lang.IndexOutOfBoundException

java.lang.ConcurrentModificationException

9.2 ERROR log messages

The ERROR log messages are used mostly for being sure that specific messages are logged.

Most of these logged errors don't stop the VDS-II from running and are more informational than critical issues.

The format of the error messages in the logs is :

*[**time**] (ERROR) [**ThreadName**] ([**ClassName**]:[**line**]) – [**Message**]*

General syntax error

Most error message are logged with the format "Exception +message", most messages should be self explanatory.

Here some examples of some of these error messages.
When the errors are coming from early phase of VDS-II starting there is a high chance that something is wrong in the configuration.

<u>ClassName :StaticRouter</u>

Message ***PARSING of routing-configuration.xml FAILED : check for xml error***
Xml error, xml parser was unable to parse correctly the file

Message : ***No routing-table configuration found! A new one has been created.***
The StaticRouter class could not find a routing-configuration.xml file a new one has been created with all channels assigned to dynamic channel pool (it needs to be configured if you want to assign specific recording to recorder channels)

Message : ***Cannot create default routing-configuration table!***
Creation of default static routing failed (check file writing rights, path etc…)

ClassName :MediaDecoder

Message : ***No decoder found for codec : XXXXXXX***
Couldn't find a decoder for the specified codec
Check mediadescription in spring-config.xml , if codec is present and/or if the payload value is correctly set (dynamic payload)

Message : ***Output decoding data empty using Alaw silence***
The output bytes from decoder is empty, alaw silence will be generated in place of the bad decoding output. Check dynamic payload value in the mediadescription or invalid received data.

Some of the old ERROR messages have been changed to INFO messages starting with version 4.x.x of the VDS-II (all routing error messages have been moved to info messages)

Routing messages

When the recorder registers to the VDS-II, it gives the VDS-II a list of channels

By default those registered channels are put into a pool of dynamic channels (any channel can be used for any recording, the number of channels = possible simultaneous recording)

When the VDS-II need a channel it acquires the first free channel from that pool.

When you want to associate specific recorder channels to specific incoming calls, it's possible to define some routing rules (called filters) which will route calls containing specific data to specific recorder channels. These rules are defined in a file called ***routing-configuration.xml.***

When that file is present every channel is assigned to a different pool of channels: dynamic, static or blocked

If a recording is failing due to routing issue some specific VDS-II routing exceptions are generated.

ChannelNotActiveException

Channel is not active (check routing-configuration if that channel is not blocked)

NoChannelAvailableException

Message : Could not acquire any channel Preferred channels [list] Free : [static] [dynamic]
Preferred channels [list] contains the list of matching channels from routing
[static] : list of free static channels
[dynamic] : list of free dynamic channels
All the channels in received list are already active or list is empty (no matching channels)

Message : No dynamic Channels available check routing-configuration file
If all your channels are assigned to static channels and/or blocked channels and the received call doesn't match any of your routing rules (check your static filters).
By default the VDS-II will try to acquire a dynamic channel when it failed acquiring a static one.
Modify static routing for matching or add dynamic/static channels.

Message : (XX) No more dynamic Channels available !!!
XX = number of dynamic channels in recording
All dynamic channels already used the new stream cannot be recorded
(Increase number of channels)

Message : Tried to acquire channel by Id for RXXXX chan X [static] [dynamic]
[static] : list of free static channels
[dynamic] : list of free dynamic channels
Trying to acquire a specific channel failed (could be blocked or already in use)

Message : Tried to acquire RXXXX chan X
Trying to acquire a specific static channel failed (could be blocked or already in use)

NoChannelStartedException

- *The channels were not started on recorder GUI or VDS-II didn't received the recorder channels*
- *Check channels status on recorder side.*
- *Check VDS-II status/registration on recorder side.*
- *Verify connection between VDS-II and recorder.*

9.3 Spring-config errors

The spring-config errors will prevent the VDS-II from starting and generate java nested stack traces leading to quite a lot of output

[2015-01-16 09:23:25,733 (ERROR) main] (Main.java:102) - Error creating bean with name 'recordingLogic' defined in file

The root cause is not that easy to read due to the amount of output.

It’s located near the end of the exception printout.

Here is a list of the most common ones

Caused by: org.springframework.beans.NotWritablePropertyException: Invalid property 'payloadType_TETRA' of bean class

The most probable error is a syntax error on a property or an invalid value.

On the above example the codec payload-type-TETRA is invalid, need to check the MediaDescription bean in the spring-config.xml

Caused by: java.net.BindException: Address already in use: JVM_Bind

Another VDS-II is already running and they have a conflicting server address or another program is already using some VDS-II ip/port combination.

You need to review your network settings/third party application configurations and/or adjust VDS-II configuration to prevent conflicts.

Caused by: java.lang.UnsatisfiedLinkError: C:\Atis\vds-eurocaertsp-active-4.0.3\voipcodecs.dll: Can't load IA 32-bit .dll on a AMD 64-bit platform

The dll are 32 bits and you are trying to run the VDS-II on a 64 bits java runtime environment.

Install a 32 bits version of the java runtime (JRE) and reinstall the VDS-II it will find the 32 bits JRE.

***** End of Document*****