

# **Installation Guide**

## **VDS-II**

## **SIP Passive**

Version 2102

---

## Table of Contents

<b>1</b>	<b>Document History.....</b>	<b>4</b>
<b>2</b>	<b>Purpose.....</b>	<b>4</b>
<b>3</b>	<b>Acronyms and Abbreviations.....</b>	<b>4</b>
<b>4</b>	<b>Introduction.....</b>	<b>4</b>
<b>5</b>	<b>Vds-II-SIP-Passive .....</b>	<b>5</b>
<b>6</b>	<b>Installation .....</b>	<b>6</b>
6.1	Installation of VDS.....	6
6.2	Network Configuration .....	15
6.3	VOX Configuration.....	16
6.3.1	Operating principle .....	17
6.3.2	Parameters description.....	17
6.3.3	Input signal quality .....	18
6.4	Routing type configuration.....	19
6.5	Verifications.....	21
6.5.1	Checking the Service status .....	21
6.5.2	Analysis of log messages .....	22
6.6	Installed Files .....	22
6.6.1	Spring-config.xml.....	22
6.6.2	Log4j.properties.....	22
6.6.3	Runtime.xml.....	23
6.6.4	Batch files.....	23
6.6.5	Dll Files.....	23
6.6.6	Wrapper files .....	23
6.6.7	Jar file.....	23
6.6.8	blacklist-filter.xml .....	23
6.6.9	whitelist-filter.xml .....	24
6.6.10	sip-port-list.xml .....	25
6.6.11	routing-configuration.xml .....	26
6.6.12	Log files .....	26
6.6.13	version.ini file.....	26
<b>7</b>	<b>VC-MDx Recorder Configuration.....</b>	<b>27</b>
7.1	Creat/Edit of ATIS VOIP Server .....	28
<b>8</b>	<b>VDS Architecture .....</b>	<b>33</b>
<b>9</b>	<b>Mapping.....</b>	<b>33</b>

---

**10 Routing.....33**

**11 Formatting.....35**

**12 Troubleshooting .....36**

    12.1 Exceptions.....36

    12.2 ERROR log messages .....37

    1. Spring-config errors.....39

## 1 Document History

Version	Date	Name	Status
1.0	2021-02-16	VR	Document created as part of VDS-114

## 2 Purpose

This document describes the installation and update of VDS-SIP-Passive software. It covers the installation and update process, the standard configuration, verifications and troubleshooting hints.

## 3 Acronyms and Abbreviations

API	Application Programming Interface
DLL	Dynamic Link Library (Microsoft)
GUI	Graphical User Interface
RTP	Real-time Transport Protocol (RFC3550)
VoIP	Voice over Internet Protocol
TETRA	Terrestrial Trunked Radio
VDS	VoIP Decoding System (VoiceCollect)
VEP	VoIP Export Protocol (VoiceCollect)
VOX	Voice Operated eXchange
SIP	Session Initiation Protocol
IP	Internet Protocol
PCAP	Packet Capture

## 4 Introduction

VDS stands for VoIP Decoding System, it is designed to transform an incoming protocol to our standard proprietary recording protocol (VoIP Export Protocol-VEP). Of course, several other features are available into the VDS if configured/enabled like Audio Mixing, Vox Activation, Transcoding and Routing.

The “VDS-II SIP Passive” supports the SIP protocol to record Air Traffic Control and Management Communications. The VDS is acting as a passive server, all recordings are piloted by the client (start recording, stop recording etc....).

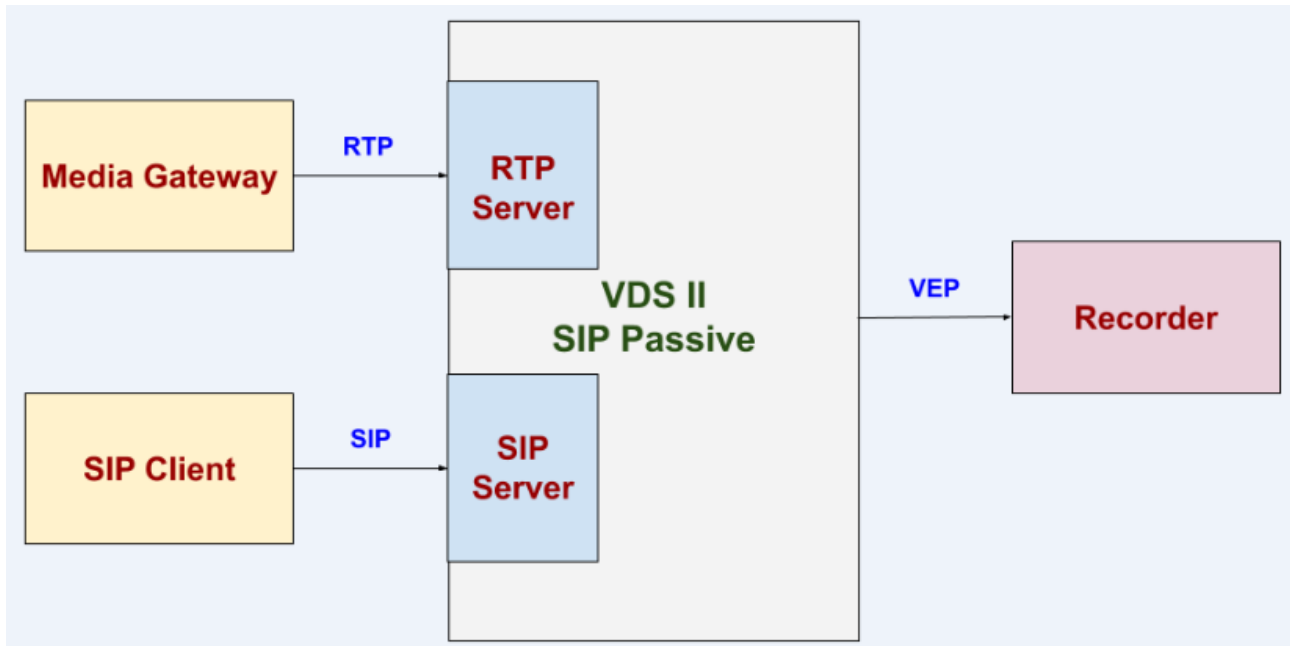
The VDS being a Java application it's portable on any hardware which can run a java runtime machine, but it has some limitation due to some codecs which are only available for windows (dll) for now (tetra,g726,g723).

VDS runs on java platform either 32bits or 64 bits 1.8 and above. It can be either Java Standard Edition or OpenJDK.

It can be started on the same machine as the recorder or on a separate machine.

Depending on the number of simultaneous recording channels and the enabled/configured features it could be recommended to start it on a separate machine.

## 5 Vds-II-SIP-Passive



The VDS:

- Handles all SIP messages
- Receives Audio through RTP listener

## 6 Installation

### 6.1 Installation of VDS

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

**NOTE:**

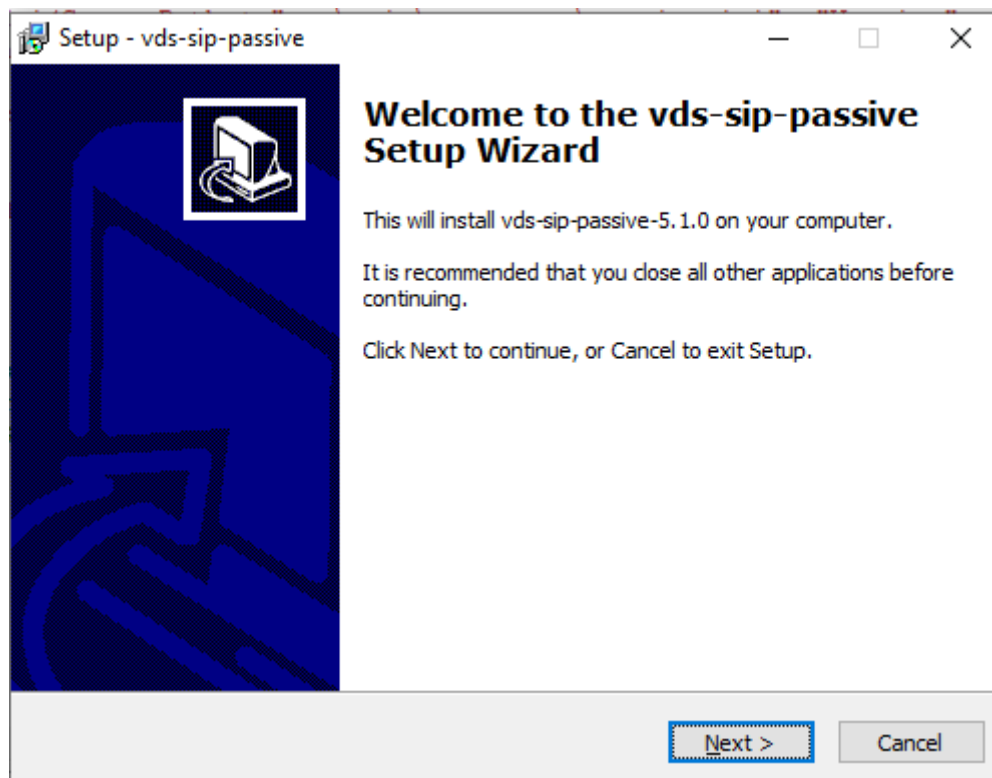
A 32-bit/64-bit version of the Java Runtime Environment with Java 1.8 or above must already be installed.

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

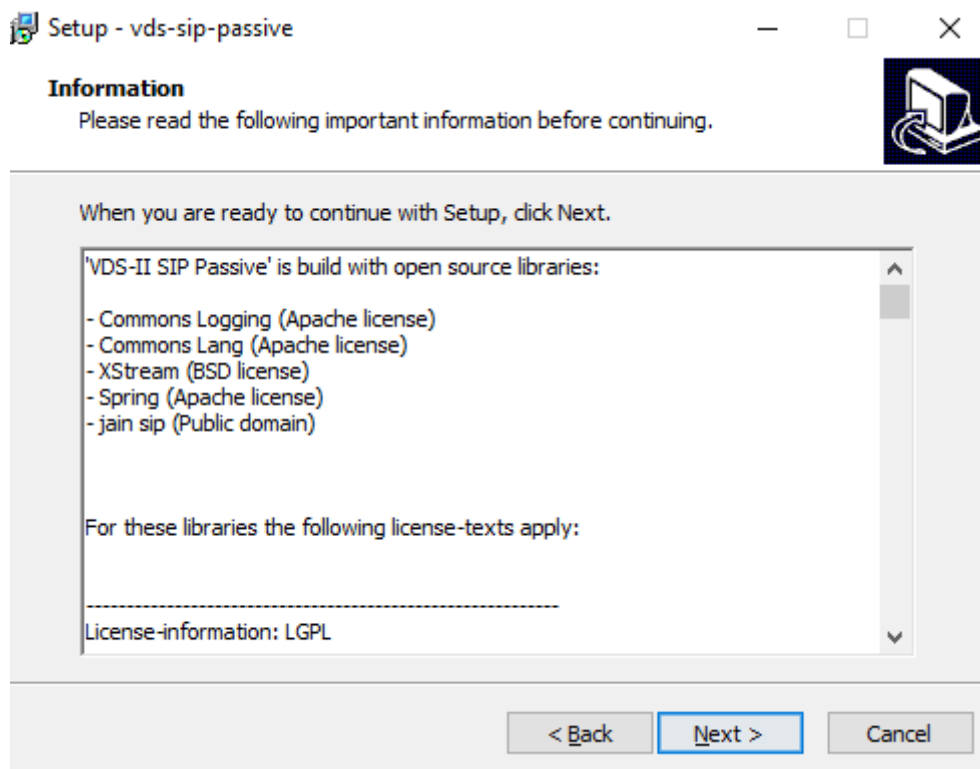
Locate the file **vds-sip-passive-x.x.x\_setup.exe** and double-click on the icon:



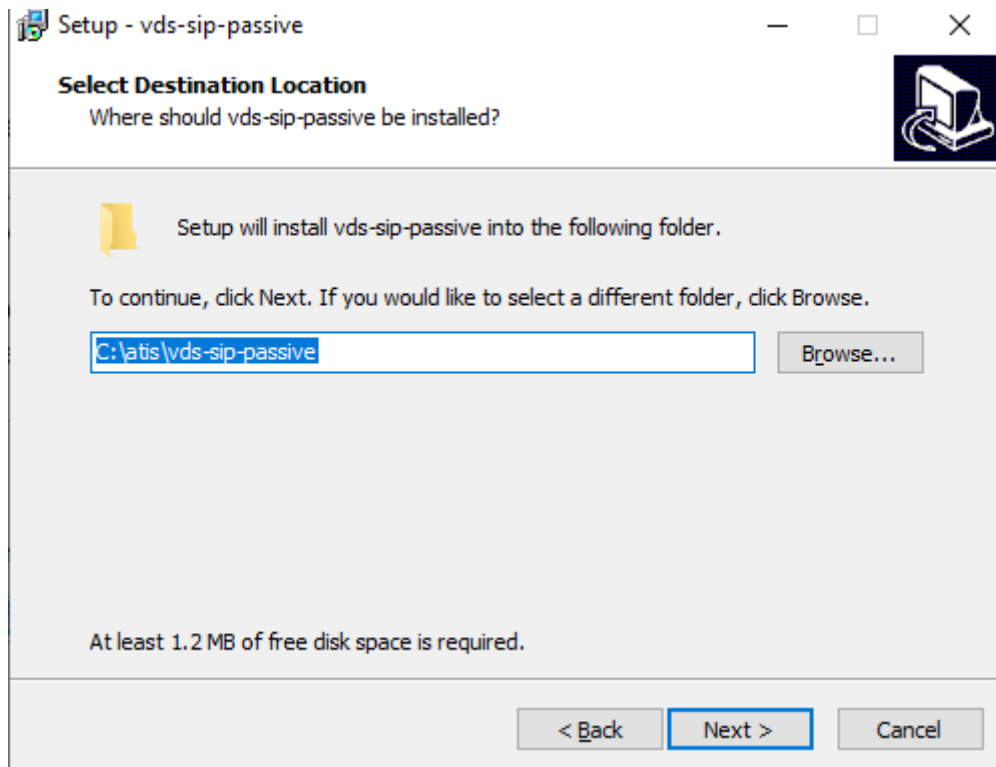
The following dialog box appears, press next:



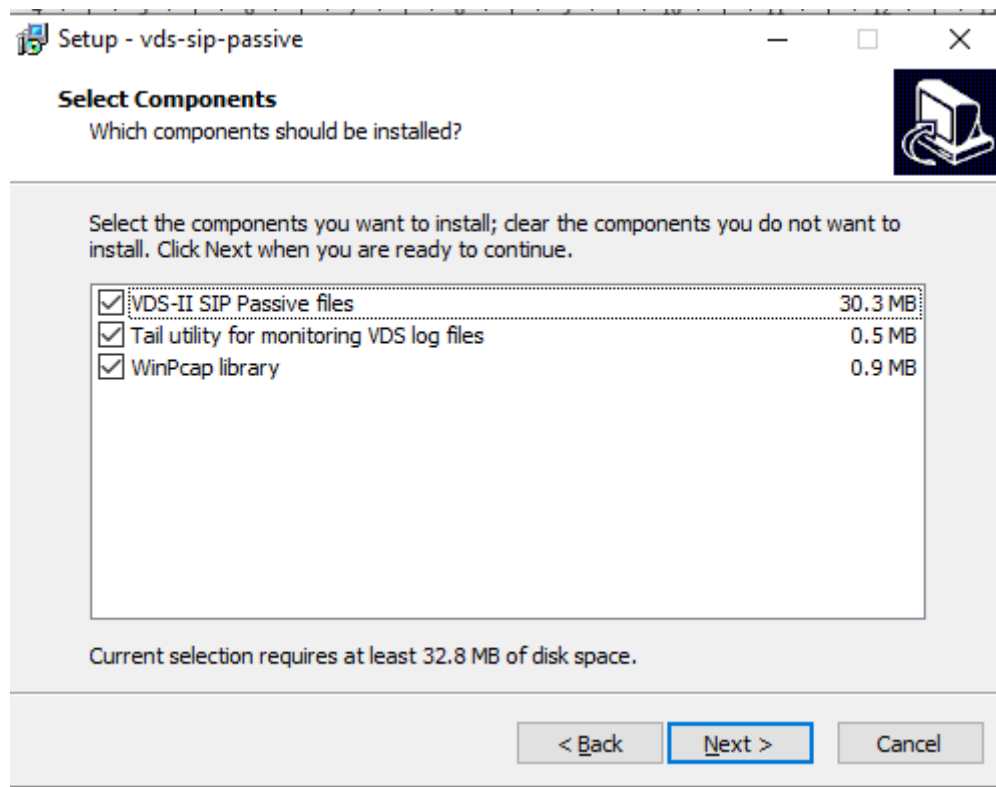
Select the Next button to proceed the installation:



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

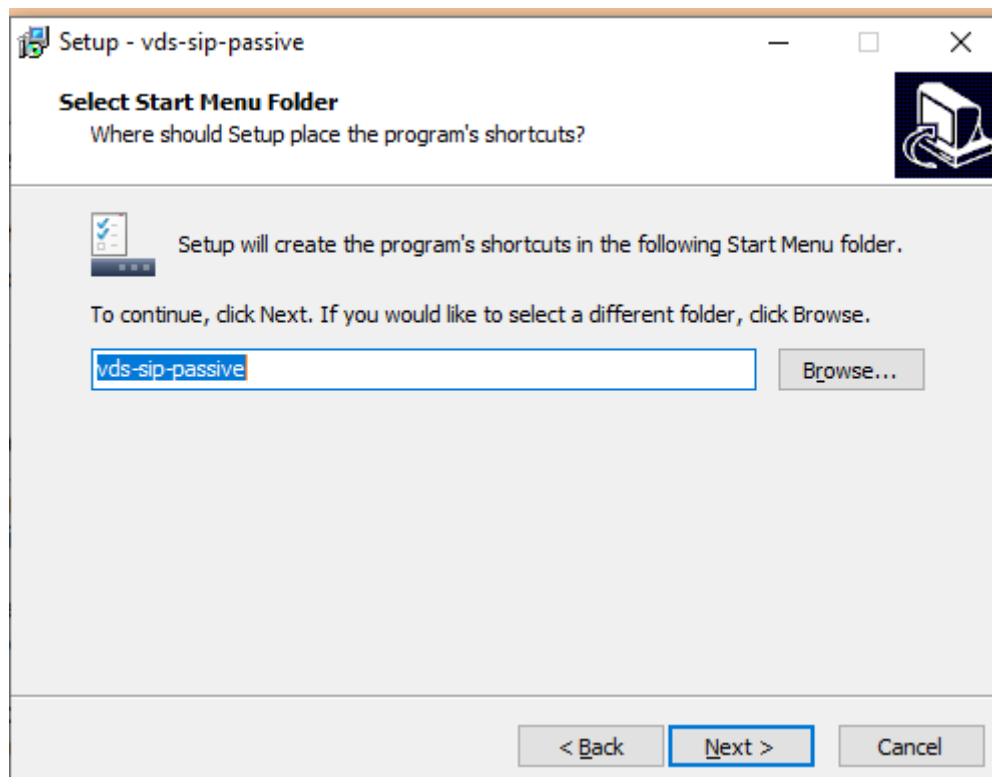


Select the desired components that is required to get installed and click Next:

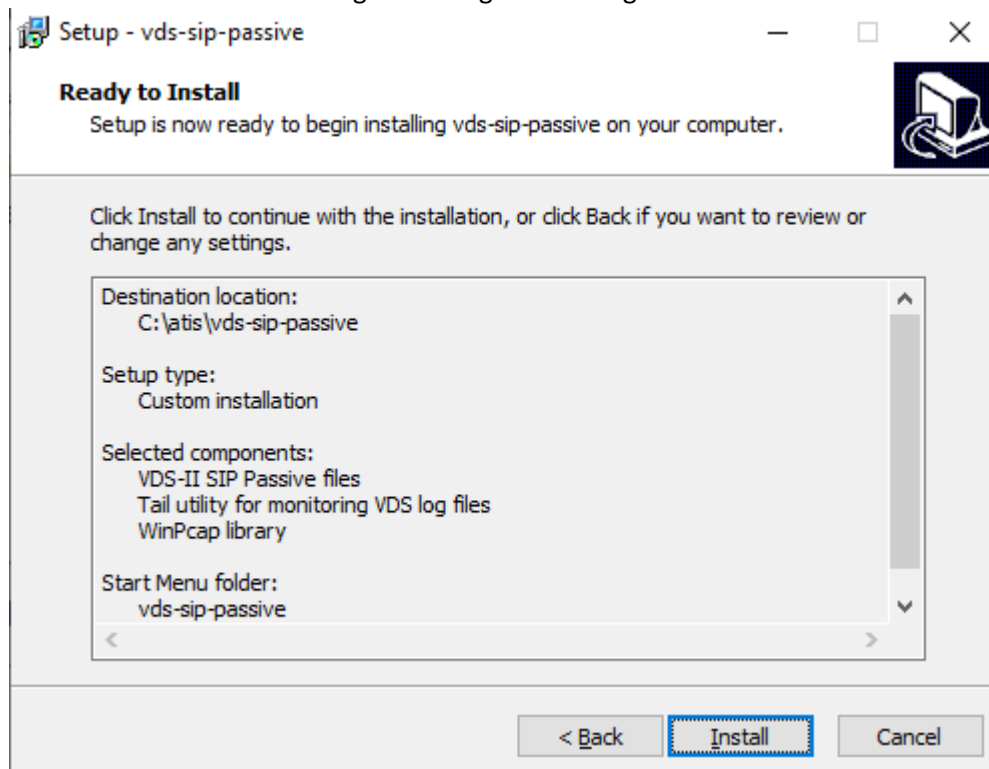




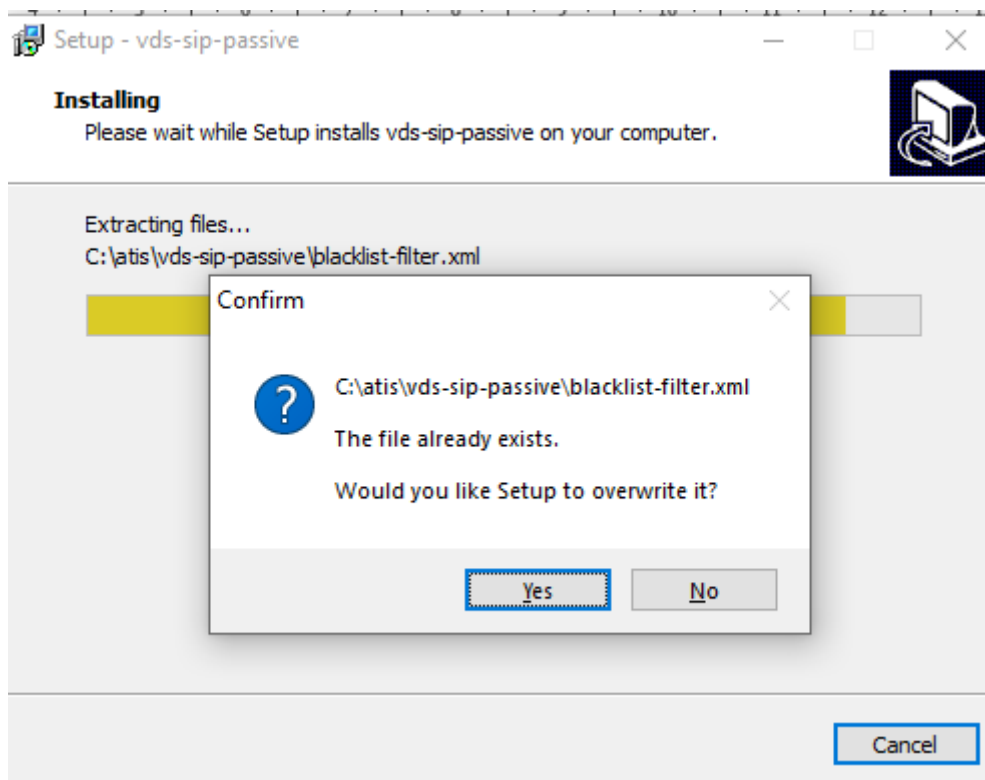
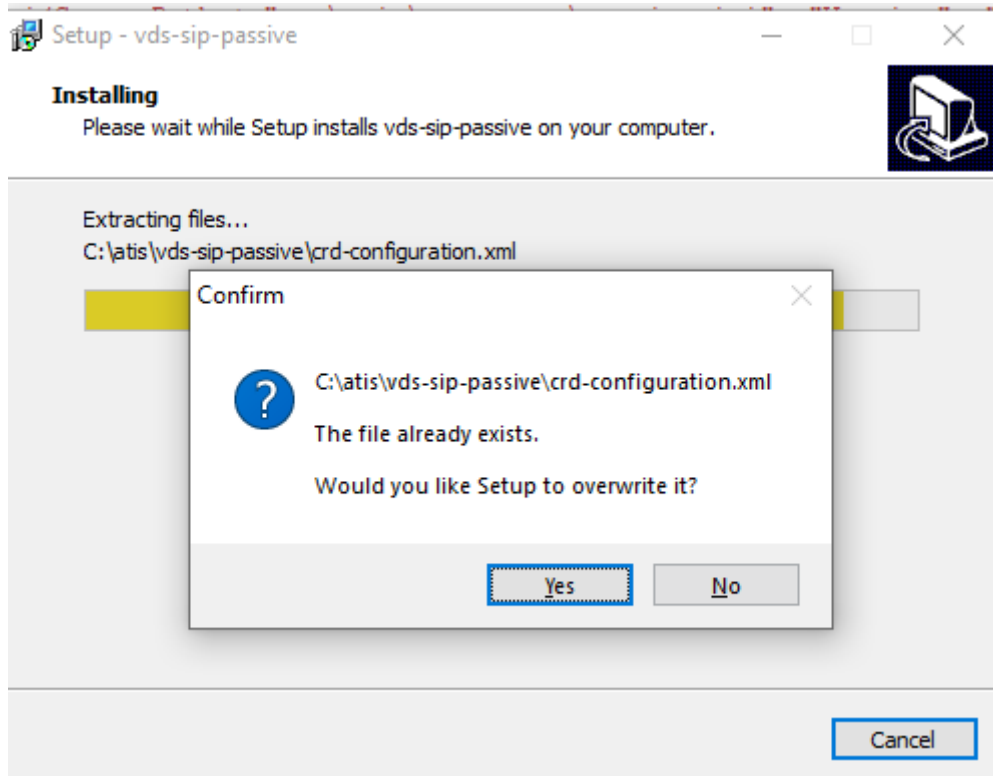
**Note:** If you are performing an update then you can safely uncheck 'WinPcap library' in this wizard as it would have been already installed during the previous installation. So that it will be not be re-installed. For fresh installation all the three components must be selected.

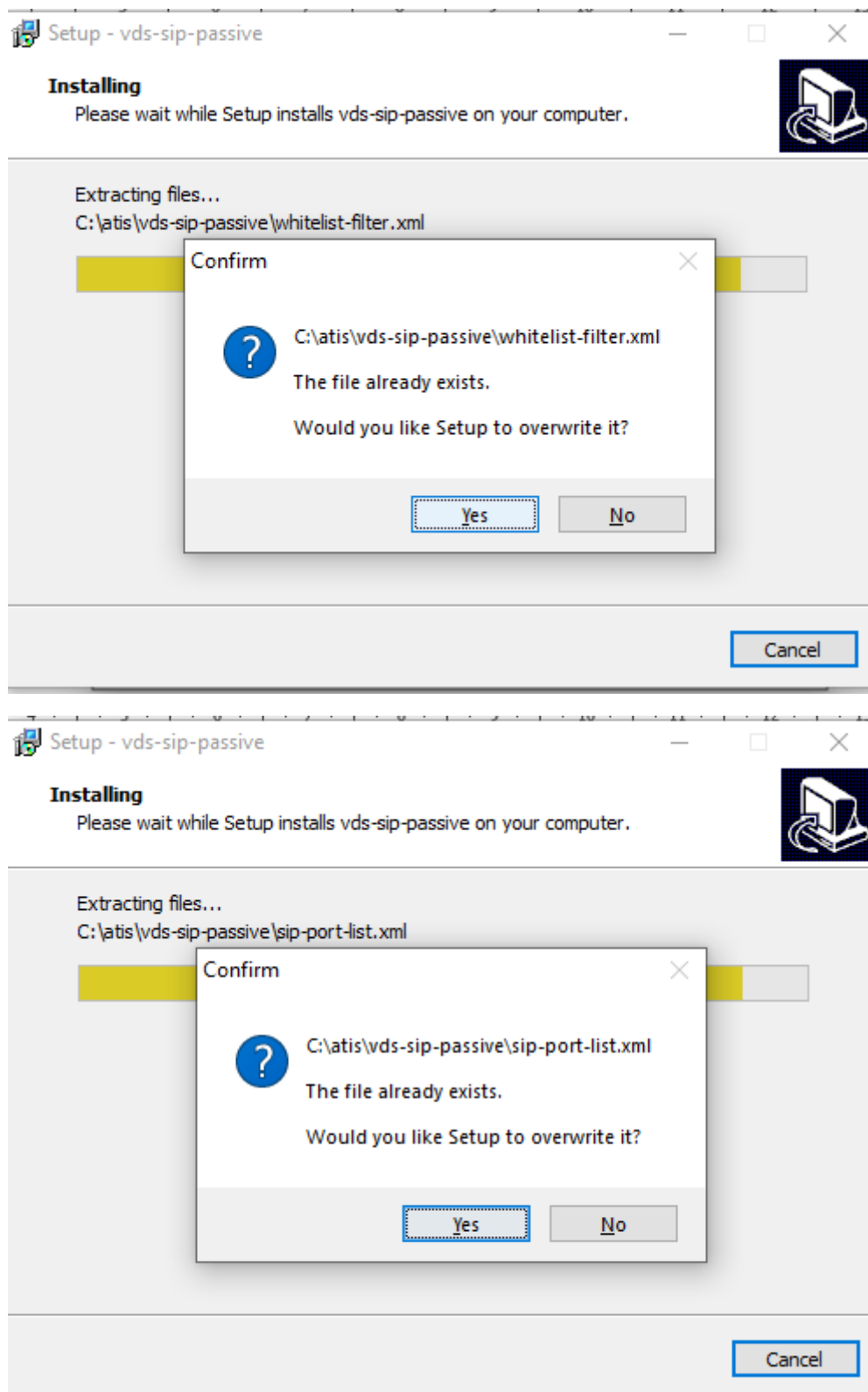


Click on Install after reviewing the configured settings.

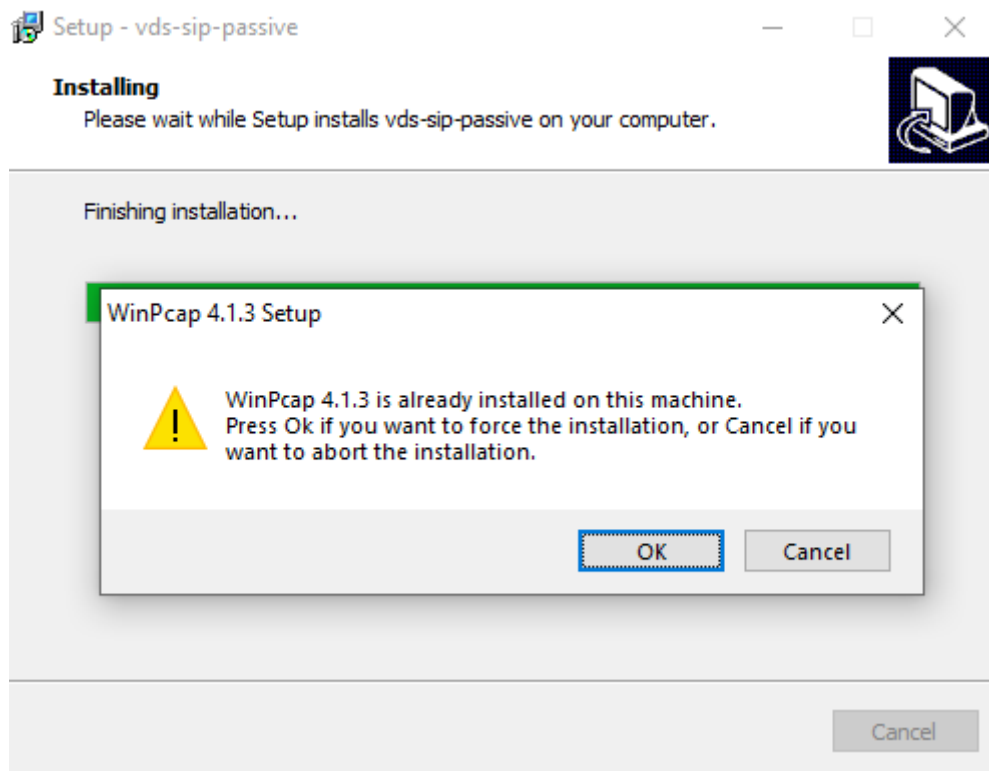


**Note:** If you are performing an update (installation in the same directory as the one of an existing VDS instance) rather than a fresh installation, you will be asked to choose whether you want to keep your configuration files or if you want to create new default ones. In most cases, you will not choose to overwrite XML configuration files.

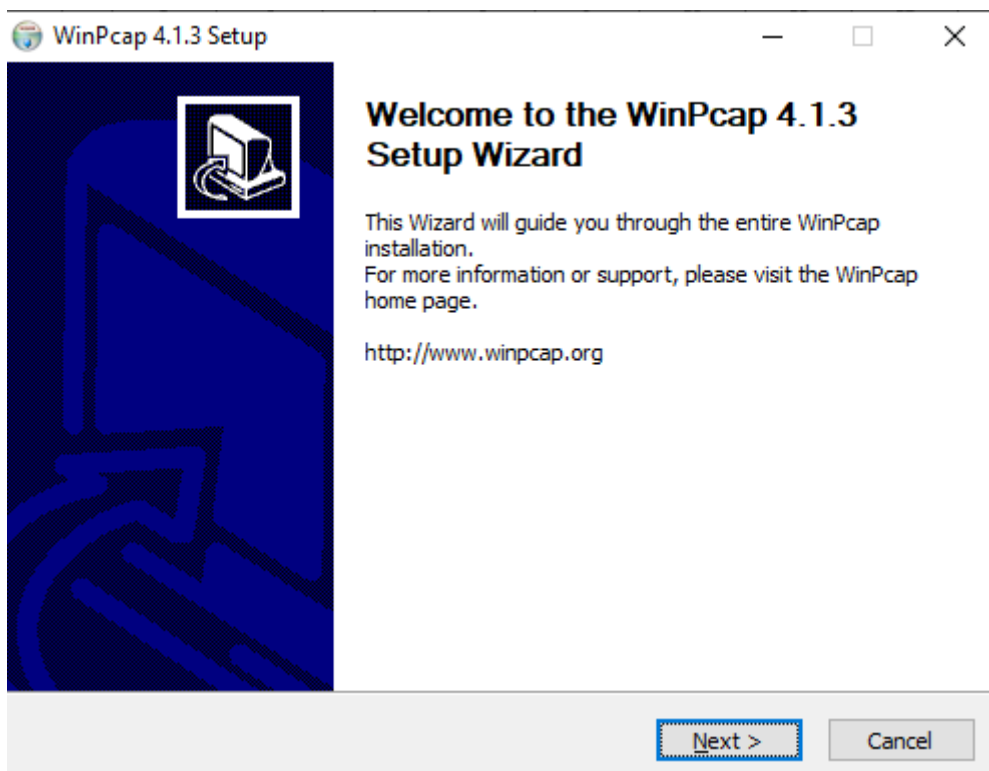


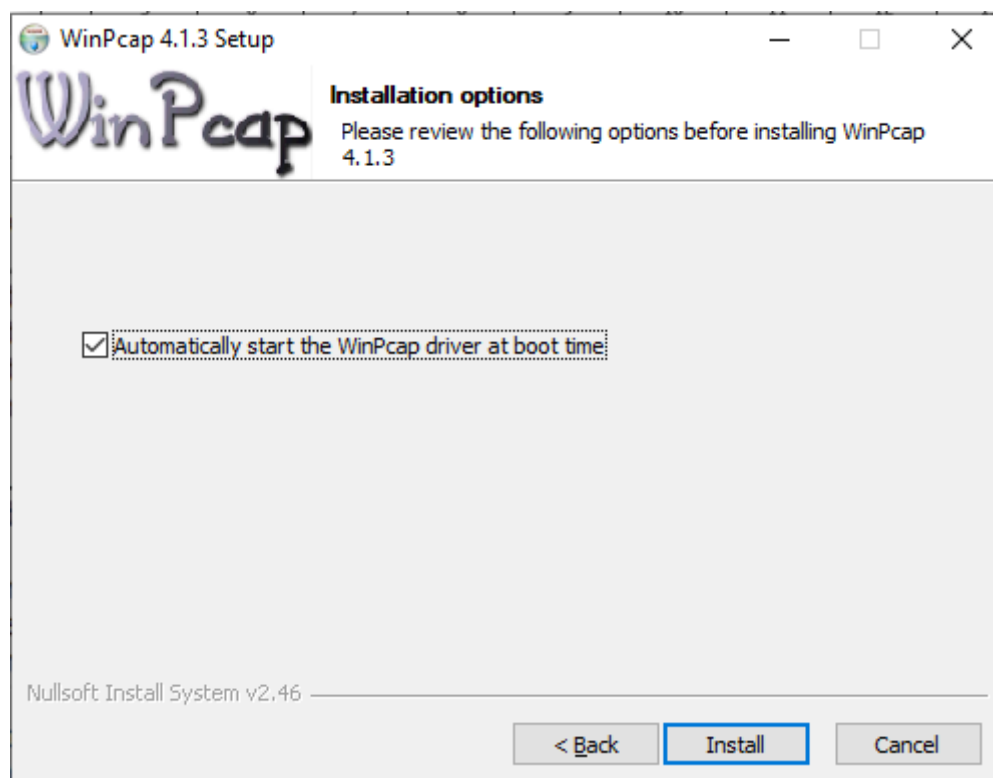
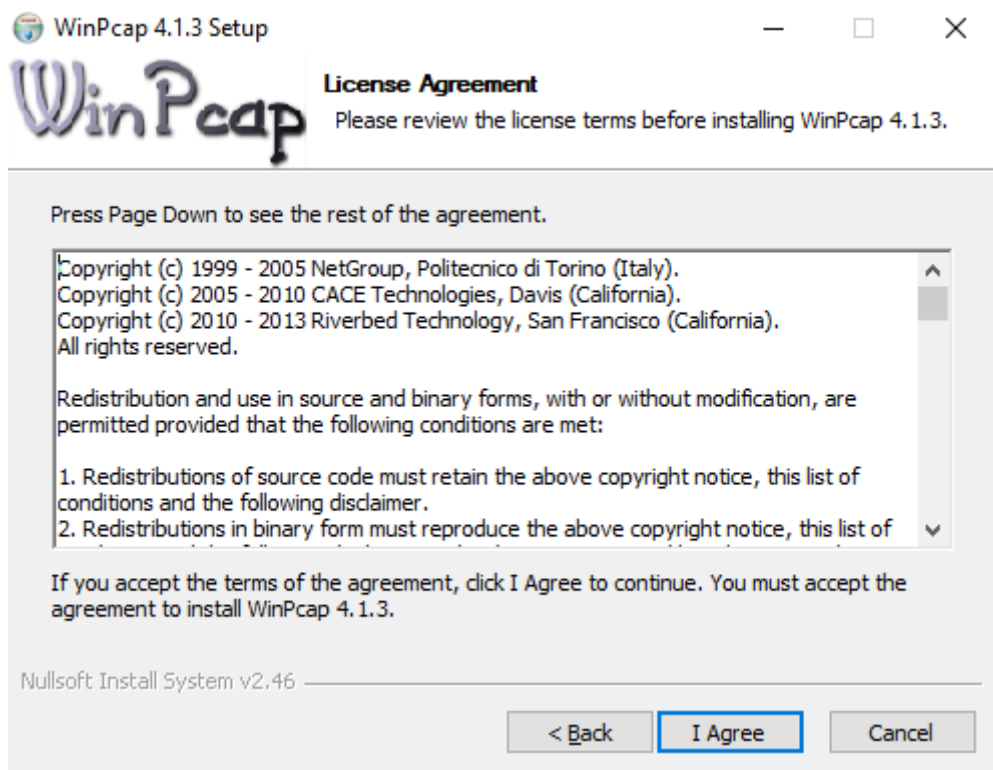


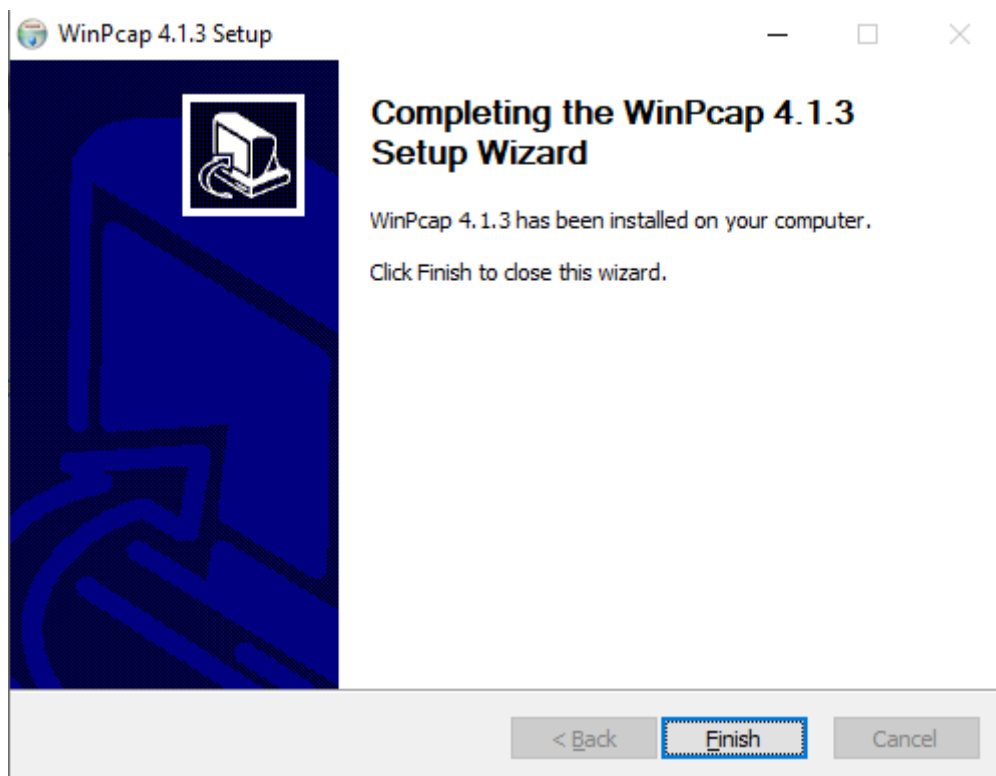
In case of Upgrade, Win PCAP library would have already installed. So the below step can be skipped by clicking on 'Cancel'.



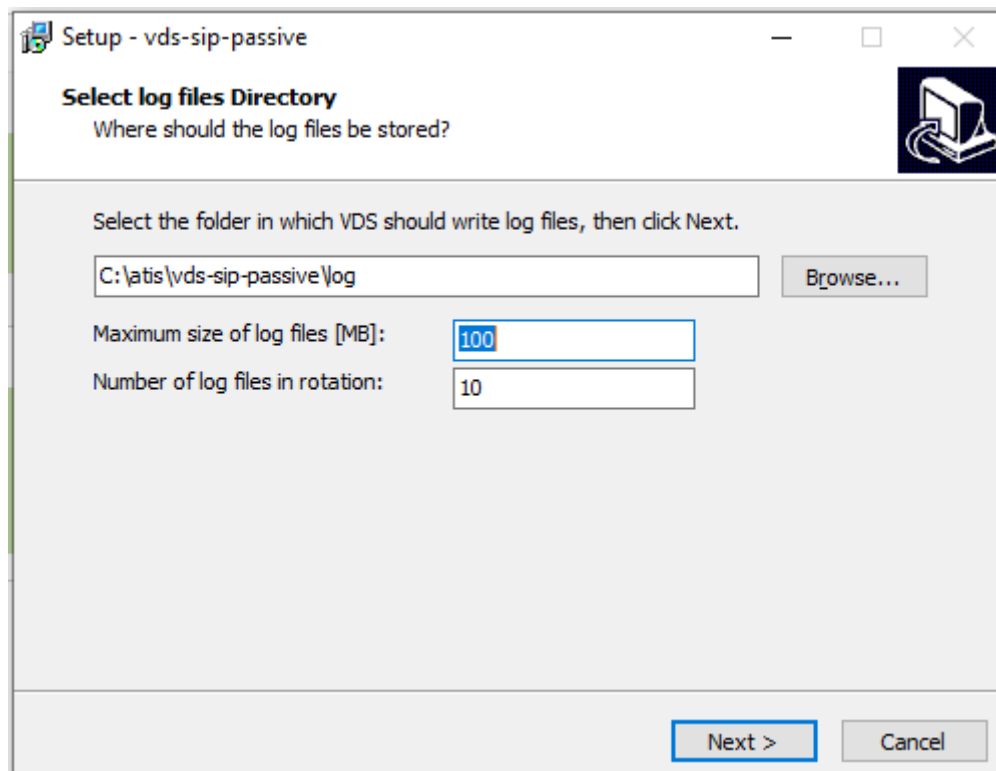
During the fresh installation, Win PCAP software will get installed at the end of VDS installation wizard. Please refer the below steps







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



## 6.2 Network Configuration

Configure the Network devices used for VoIP Packet capture, IP address of network interface for VEP connection with Recorder and the type of filtering from the below wizard.

Setup - vds-sip-passive

**VDS-II SIP Passive - Network Configuration**

Please select the network interfaces used and the type of the IP-filter  
Then click Next.

Network devices used for VoIP packet capture:

<input type="checkbox"/>	1 - Microsoft	192.168.166.144
<input type="checkbox"/>	2 - Orade	192.168.56.1

IP address of network interface for VEP connection with Recorder (0.0.0.0 => Any):

0.0.0.0

Select the type of filtering for unwanted network traffic:

☒ Blacklist-based filtering

☐ Whitelist-based filtering

< Back   Next >   Cancel

Parameters:

- **Network devices used for VoIP Capture:** Select the desired network interfaces of IP addresses to be listened by VDS
- **VEP IP:** IP Address of the network interface connected to VC-MDx.

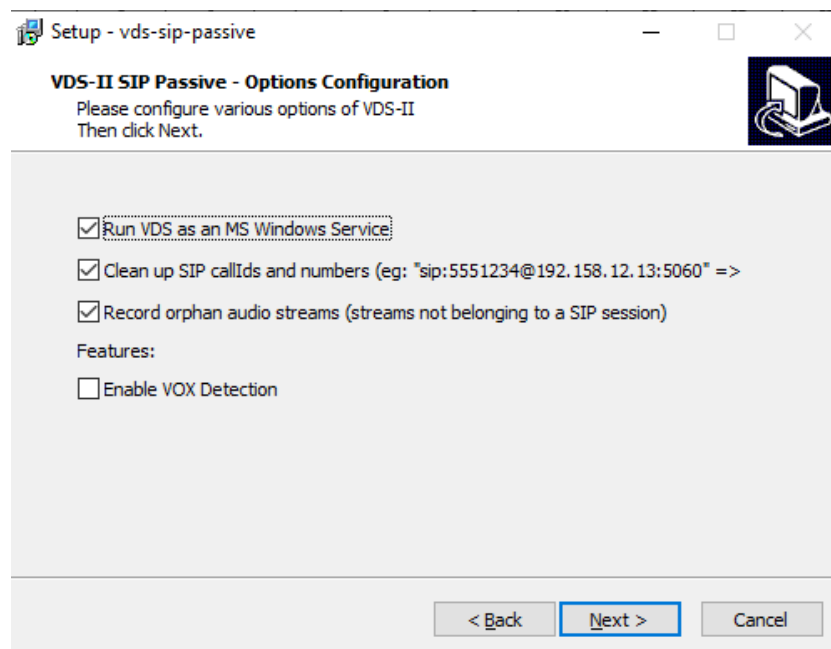
**Note:** In the situation of a software update, the values will not automatically be set to the old ones. To set the same values as previously, you can open the old **spring-config.xml** file and retrieve the values that must be entered by looking at the 'theloModule' bean definition.

**Note:** Both Ipv4 and Ipv6 addresses can be selected.

[Refer section 6.6.8 for Blacklist based filtering](#)

[Refer section 6.6.9 for Whitelist based filtering](#)

After the desired configuration, Click on Next



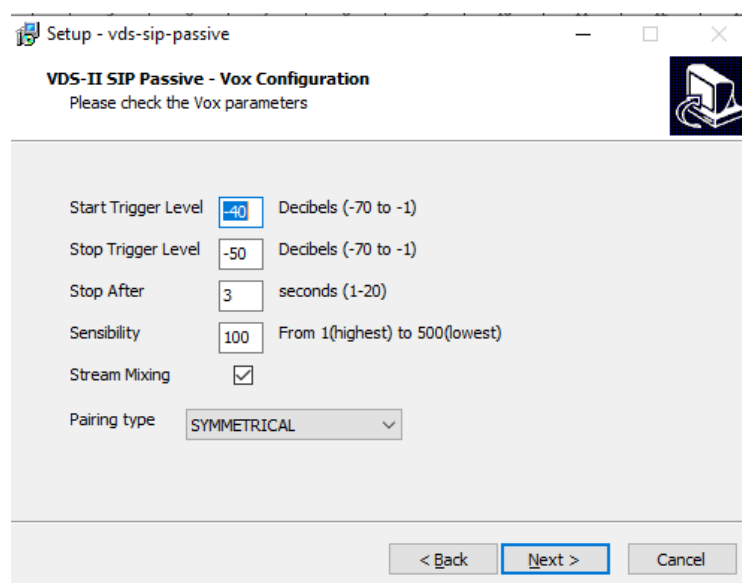
Run VDS as a service (checkbox): By default, we recommend to run VDS as a service. If it is not checked the VDS will be run in console mode (be cautious any click in the console will freeze the VDS).

Enabling '**Clean up SIP callIds and numbers**' configuration results in the creation of formatting-configuration.xml file to clean up the SIP addresses for the fields such as **aParty**, **bParty** and **callId** to only show the call numbers

Enabling '**Record orphan audio streams**' configuration would make the VDS to record the stand alone RTP sessions (Without SIP). In case if this configuration is disabled then the VDS would record only the SIP + RTP Sessions.

By Default the VOX configuration will be disabled. To activate VOX, enable VOX Detection in this wizard and the relevant configurations can be done in this interface. [Refer 6.3 – VOX Configuration](#)

## 6.3 VOX Configuration

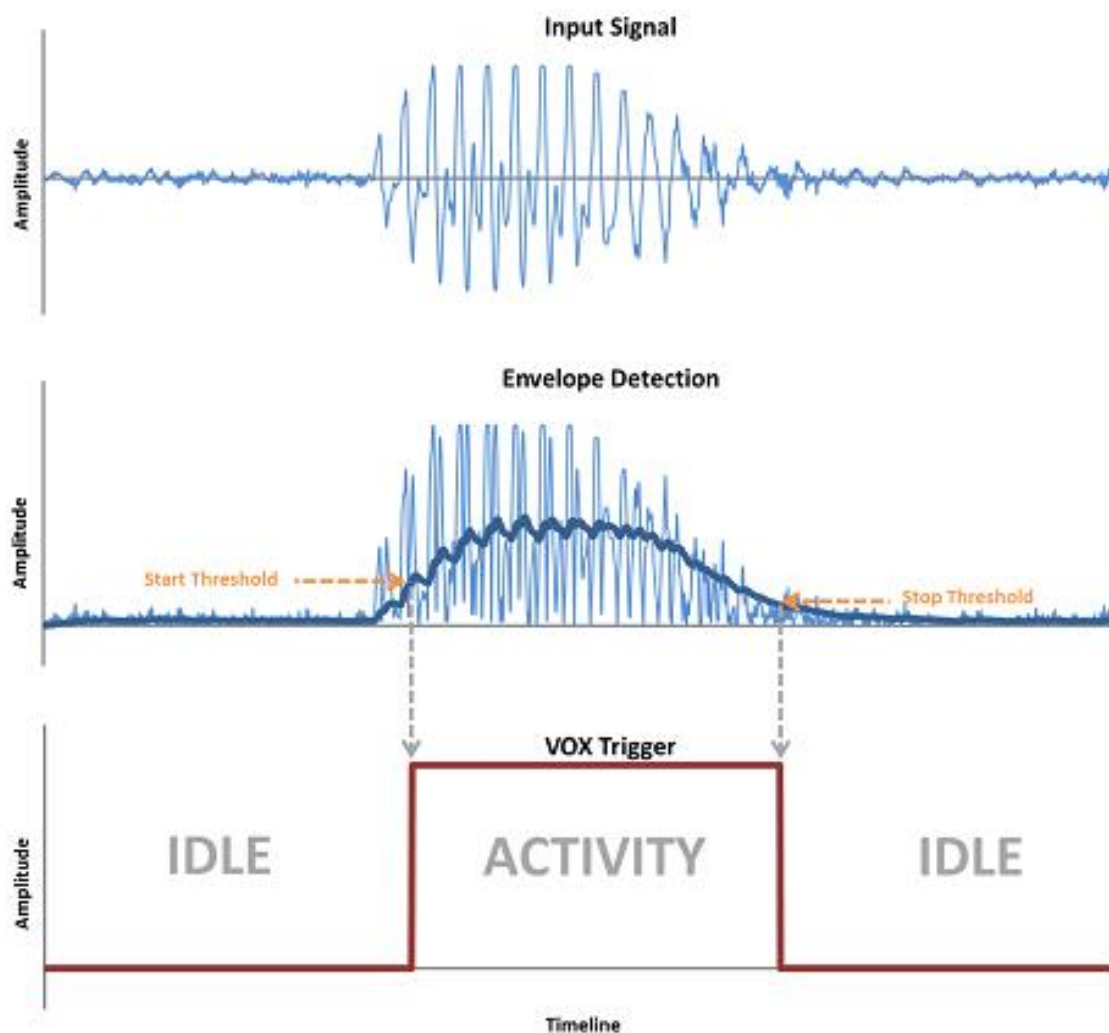




VDS-II SIP Passive supports a VOX feature which allows for producing records triggered by the activity (voice) at the input. Activation of this feature requires the configuration of several VOX parameters, in particular the signal envelope detection thresholds.

### 6.3.1 Operating principle

A full-wave rectification is performed on the input waveform which is then smoothed by way of a low-pass filter to detect the signal envelope. Input activity is detected when the envelope level passes over a start threshold while silence is detected when the envelope level passes below a stop threshold which is set lower than the start threshold (Schmitt Trigger).



Note: when silence is detected, VDS will wait an additional "hang" time before to actually stop current recording. This allows to keep recording during brief speech pauses.

### 6.3.2 Parameters description

VOX will work with default values in most cases. In case if you need to adjust some VOX parameters you can configure them in the wizard for the correspondig parameters.

Do refer the description of each parameter as shown below and tune them accordingly.

VOX Property	Description	Range of Values	Units	Remarks
Start Trigger Level	Trigger level for activity detection	-70 to -1 [Default: -40]	dB ( Decibel )	Must be set higher than 'Stop Trigger Level'
Stop Trigger Level	Trigger level for silence detection	-70 to -1 [Default: -50]	dB ( Decibel )	Must be set lower than 'Start Trigger Level'
Stop After	Hang time for stop detection	1 to 20 [Default: 3]	Sec ( Seconds )	Recording will be stopped only if signal stays under 'Stop After' for that time. Allows for short pauses during recording
Sensibility	Degree of smoothing in signal envelop detector	1 to 500 [Default: 100]		The higher, the less sensible to changes [If too many starts get triggered by short audio cracks, increase this value] Values too high decrease responsiveness
Stream Mixing	It determines how the caller and callee streams to be recorded	[Default: Enabled]		If it is enabled the both will be in same stream and if it is disabled then each will be in different stream
Pairing Type	Caller and Callee Stream port sameness for recording	[Default: SYMMETRICAL]		SYMMETRICAL: Caller and Callee streams can be in different port  ASYMMETRICAL: Caller and Callee streams should be in same port.

### 6.3.3 Input signal quality

In order to ensure proper VOX detection, good quality of the input signal is essential.

Ensure the gain structure (the gain adjustment of the various elements in the audio equipment chain) is correct.

Do not amplify too much in one stage and then attenuate too much in following one. Ideally, most stages should be close to unity gain (0dB. No amplification, no attenuation) and one early stage in the chain should be set to take advantage of maximum available dynamic range.

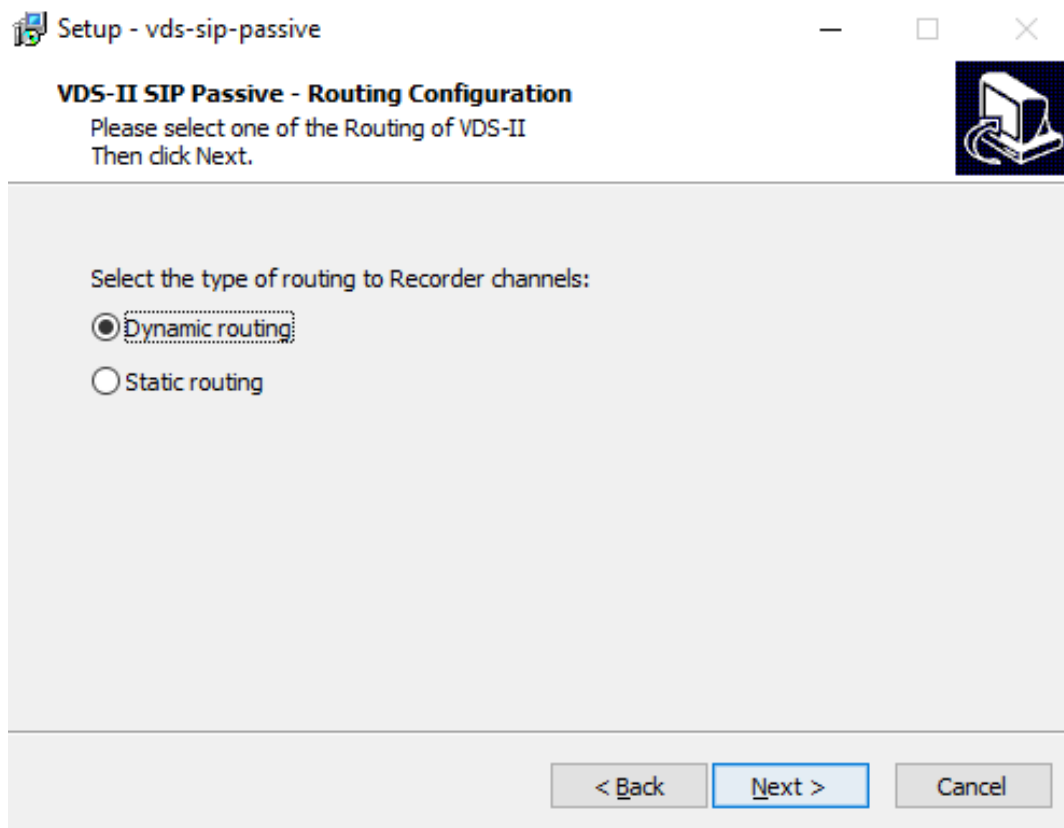
An important stage for a VoIP system is the conversion from analog to digital. The maximum dynamic range below clipping (max headroom) for a linear 16-bit PCM signal is the full scale value 32167, which corresponds to 0dB FS. A gain structure that is correctly set should provide VoIP audio with optimal PCM amplitude close to 0dB FS and best possible signal to noise ratio.

Be prudent with AGC (Automatic Gain Control) features because they will amplify the noise too and if the input signal is weak, the VOX feature will not make the difference between voice and noise. A squelch feature before the AGC would help in this case.

## 6.4 Routing type configuration

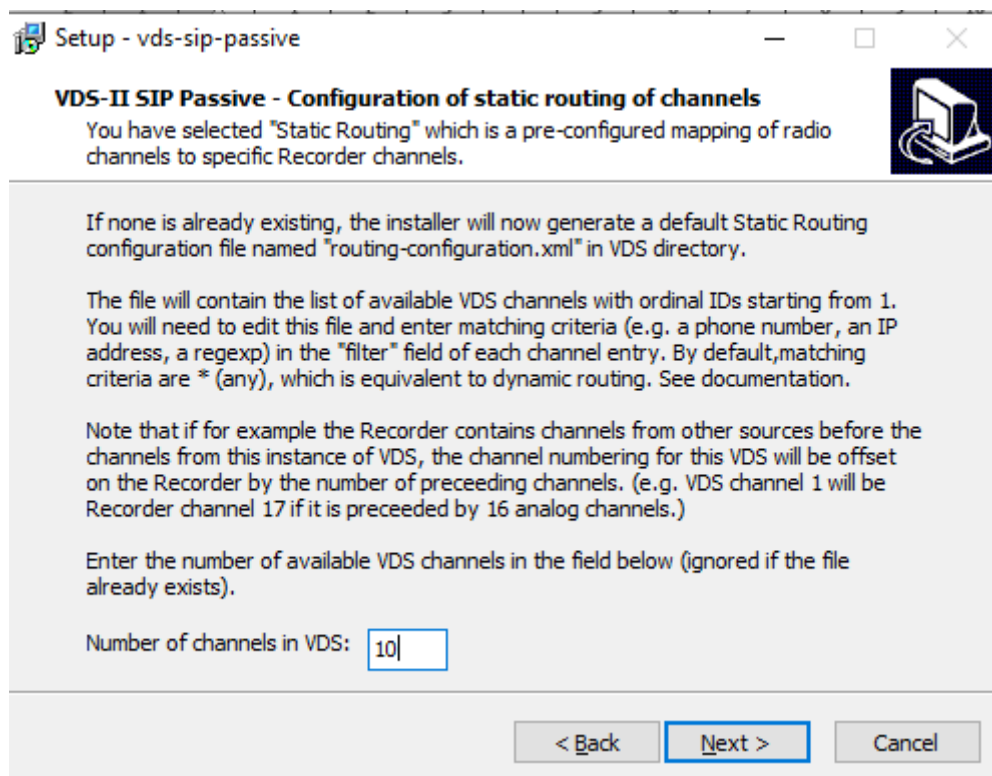
In the next page, you will need to select the type of routing you like to use. If you are not sure what is the routing strategy of your system, please refer to the **Advanced Configuration Guide**.

In case of a software update, the installer will highlight the routing type found in your previous settings.



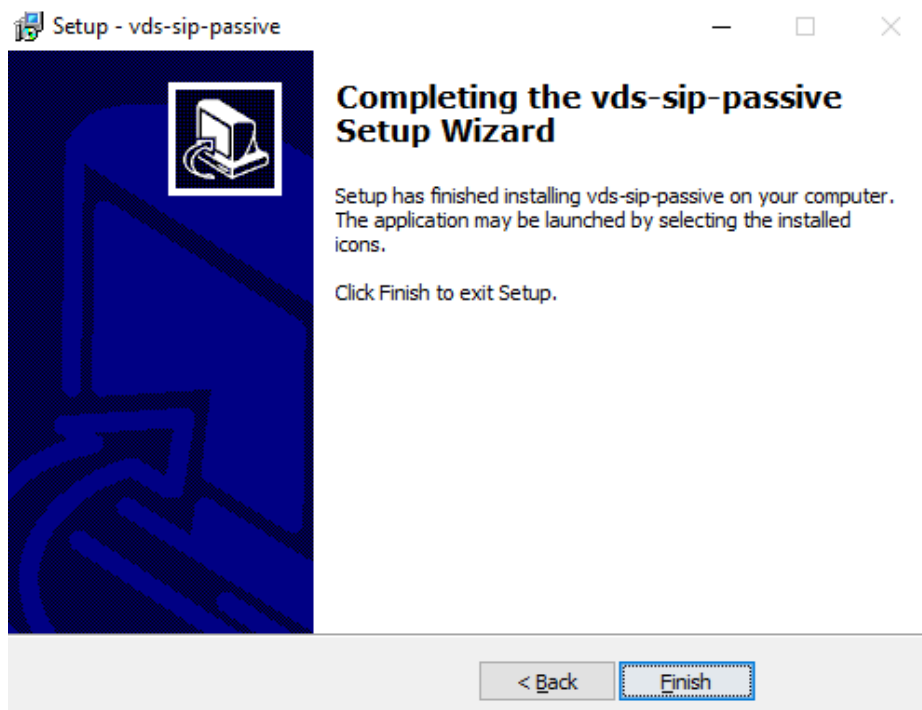
### Note:

If you choose Static routing then the following wizard will be prompted to enter the number of channels. The routing-configuration.xml will be populated in the installation directory with default configuration.



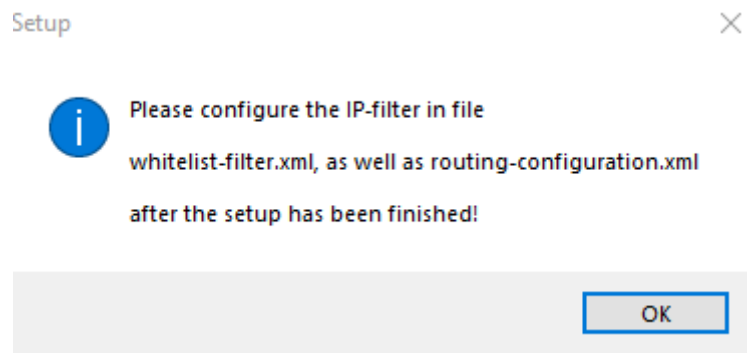
[Refer Section 6.6.11 for more details.](#)

Click on Finish the Setup to close the Installation Wizard.

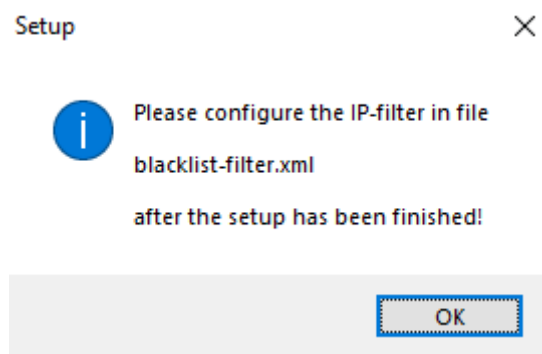


**Note:**

The sample prompt for the configuration with whitelist-filter.xml option and static routing selection.

**Note:**

The sample prompt for the configuration with blacklist-filter.xml option and dynamic routing selection.

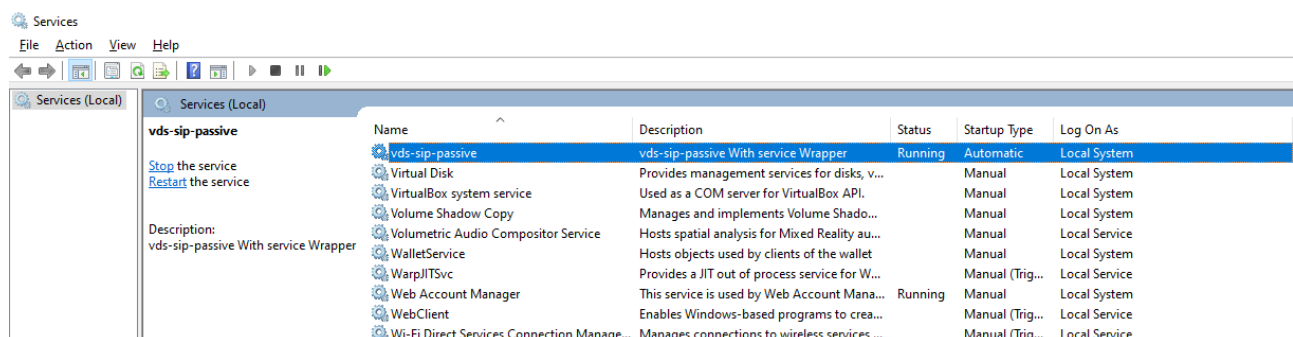


## 6.5 Verifications

### 6.5.1 Checking the Service status

After installation check With MS Windows Service Manager, you can verify that the VDS service has been started:












Note: This should be done only when 'Run VDS as an MS Windows Service option' is checked during the installation.



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

## 6.5.2 Analysis of log messages

**Run-SnakeTail** is a program located in the installation directory that offers a very convenient way of viewing and analyzing the application log messages.

PC > Windows (C:) > atis > vds-sip-passive			
Name	Date modified	Type	Size
 start-vds-sip-passive-service.bat	05-02-2021 12:34	Windows Batch File	1 KB
 stop-vds-sip-passive-service.bat	05-02-2021 12:34	Windows Batch File	1 KB
 install-vds-sip-passive-service.bat	05-02-2021 12:34	Windows Batch File	1 KB
 remove-vds-sip-passive-service.bat	05-02-2021 12:34	Windows Batch File	1 KB
 Run-SnakeTail	05-02-2021 12:34	Shortcut	1 KB
 vds-sip-passive-5.1.0-jar-with-dependencies.jar	03-02-2021 13:36	JAR File	29,857 KB
 version.ini	03-02-2021 13:36	Configuration sett...	1 KB
 tail.xml	20-01-2021 10:34	XML File	3 KB
 WinPcap_4_1_3.exe	20-01-2021 10:34	Application	894 KB
 wrapper.jar	20-01-2021 10:34	JAR File	119 KB
 atis_auge_32.ico	20-01-2021 10:34	ICO File	9 KB

Double-click on the **Run-SnakeTail**. A new Window will open and display the logs messages as they are generated. It is powered with fully customizable settings such as Font, Color, Keyword/Regular expression-based data highlighting. This can be configured in View Options under Edit Menu of the application.

The default messages font color scheme is:

- **RED**: Error
- **Yellow**: Warning
- **Light green**: Info
- **Dark green**: Debug

## 6.6 Installed Files

### 6.6.1 Spring-config.xml

This file describes the VDS 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 does not 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 VoiceCollect.

### 6.6.2 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 (it can take up to 30 seconds for the changes to become active).

### 6.6.3 Runtime.xml

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

### 6.6.4 Batch files

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

- install-vds-sip-passive-service.bat - Install the service
- remove-vds-sip-passive-service.bat - remove the service

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

### 6.6.5 Dll Files

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

### 6.6.6 Wrapper files

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

### 6.6.7 Jar file

This is the main VDS software

### 6.6.8 blacklist-filter.xml

Configure to block the Payload data from specific IP address and ports to VDS application. Below are the sample configurations to perform the same.

<!--

*This file is used to setup the Blacklist. For the setup we distinguish between three different address-types.*

1. Type: "all"

*If an IP-address has the type "all" then an IP packet is ignored when the source-address OR the destination-address matches with this address.*

2. Type: "source"

*If an IP-address has the type "source" then an IP packet is ignored when the source-address matches with this address.*

3. Type: "destination"

*If an IP-address has the type "destination" then an IP packet is ignored when the destination-address matches with this address.*

*IP-addresses can be defined as a 'single' IP address or as an range. For example:*

*Single IP address:   "192.168.1.1"*

Range of IP addresses: "192.168.1.1-192.168.1.255"

Port-Numbers can be defined as 'single' port number or as range. For example:

Single Port-Number: "5060"

Range of port-numbers: "5060-5070"

Either for IP-address or port-number a wildcard '\*' can be used too. It means that the IP-address or port-number is not evaluated for the blacklist-filter.

The IP-address and the port-number must be separated by a colon.

Here some examples:

```
<entry key="192.168.1.1:1000"          value="all" />
<entry key="192.168.2.1:2000"          value="source" />
<entry key="192.168.3.1:3000"          value="destination" />

<entry key="192.168.1.1:1000-2000"      value="destination" />
<entry key="192.168.1.1-192.168.1.255:1000" value="destination" />
<entry key="192.168.1.1-192.168.1.255:1000-2000" value="destination" />
<entry key="192.168.1.1:*"              value="destination" />
<entry key="192.168.1.1-192.168.1.255:*" value="destination" />

<entry key="*:1000"                    value="destination" />
<entry key="*:1000-2000"                value="destination" />
```

-->

## 6.6.9 whitelist-filter.xml

Configure to allow the Payload data from specific IP address and ports to VDS application. Below are the sample configurations to perform the same.

<!--

*This file is used to setup the Whitelist. For the setup we distinguish between three different address-types.*

1. Type: "all"

*If an IP-address has the type "all" then an IP packet is recorded when the source-address OR the destination-address matches with this address.*

2. Type: "source"

*If an IP-address has the type "source" then an IP packet is recorded when the source-address matches with this address.*

3. Type: "destination"



If an IP-address has the type "destination" then an IP packet is recorded when the destination-address matches with this address.

IP-addresses can be defined as a 'single' IP address or as an range. For example:

Single IP address: "192.168.1.1"

Range of IP addresses: "192.168.1.1-192.168.1.255"

Port-Numbers can be defined as 'single' port number or as range. For example:

Single Port-Number: "5060"

Range of port-numbers: "5060-5070"

Either for IP-address or port-number a wildcard '\*' can be used too. It means that the IP-address or port-number is not evaluated for the whitelist-filter.

The IP-address and the port-number must be separated by a colon.

Here some examples:

```
<entry key="192.168.1.1:1000" value="all" />
<entry key="192.168.2.1:2000" value="source" />
<entry key="192.168.3.1:3000" value="destination" />

<entry key="192.168.1.1:1000-2000" value="destination" />
<entry key="192.168.1.1-192.168.1.255:1000" value="destination" />
<entry key="192.168.1.1-192.168.1.255:1000-2000" value="destination" />
<entry key="192.168.1.1:*" value="destination" />
<entry key="192.168.1.1-192.168.1.255:*" value="destination" />

<entry key="*:1000" value="destination" />
<entry key="*:1000-2000" value="destination" />
```

-->

## 6.6.10 sip-port-list.xml

This file contains the list of ports which will be used for sniffing SIP signalling. Default configuration will be 5060 (standard sip port) and 5061 (encrypted SIP port) is shown below.

-->

```
<util:set id="sip-port-list" set-class="java.util.HashSet">
    <value type="int">5060</value>
    <value type="int">5080</value>
</util:set>
```

### 6.6.11 routing-configuration.xml

In case of fresh installation new file will be created. During the upgrade it would prompt to retain the older configuration file.

The file will contain the list of available VDS channels with ID's starting from 1.

[Refer Section 10 for more details.](#)

### 6.6.12 Log files

Two log files are created into the directory chosen during installation

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

### 6.6.13 version.ini file

It contains the below contents

**[Version]**

**Jar=5.0.10**

**Compiler=8**

The JAR value determines the VDS Application version and Compiler version determines under which this Java application compiled

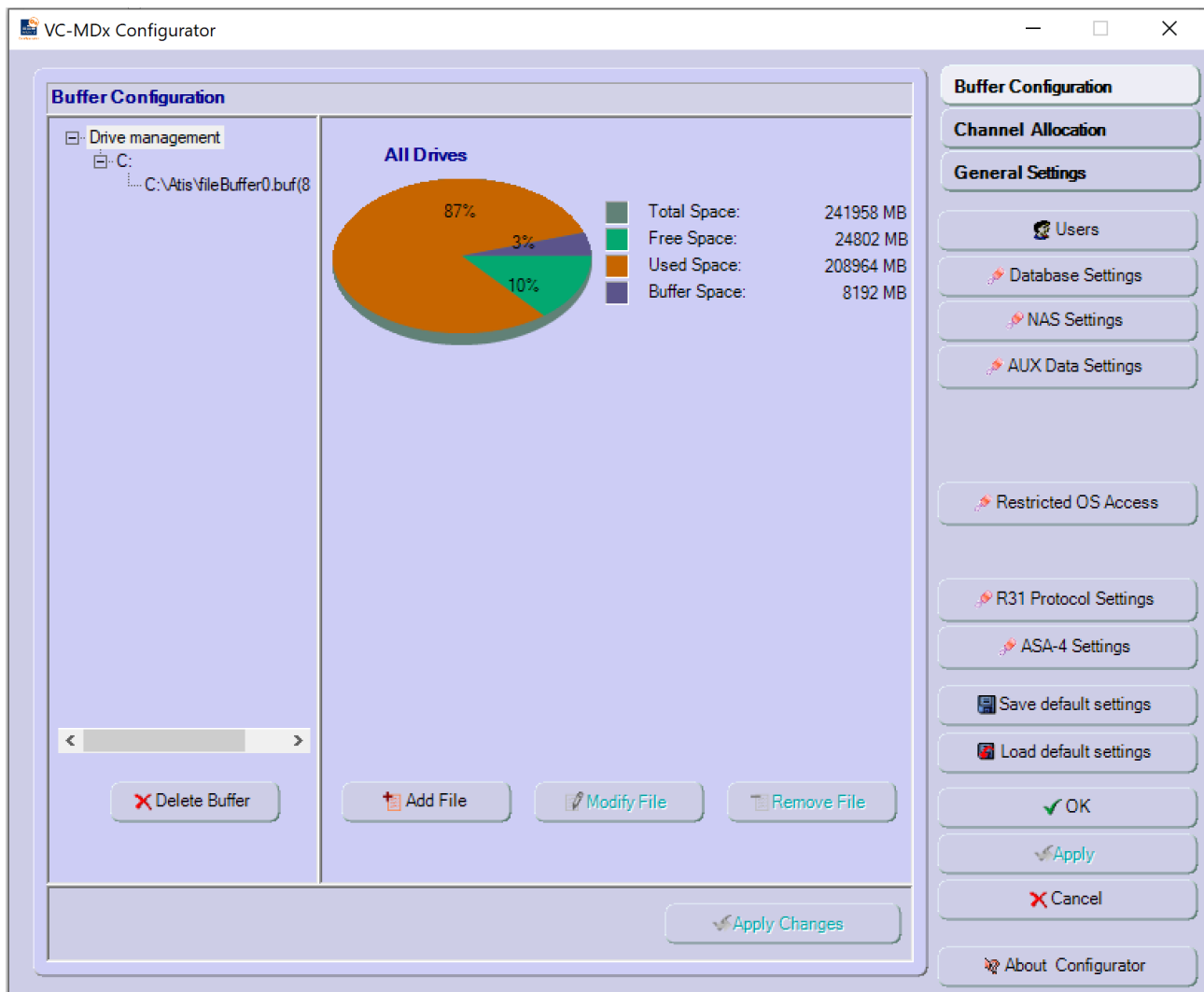
## 7 VC-MDx Recorder Configuration

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

**Note:** During the configuration of the Recorder, the VDS application must be running on the interface computer.



After authentication, the following dialog box appears:



## 7.1 Creat/Edit of ATIS VOIP Server

Click on the button '**Channel Allocation**'

VC-MDx Configurator

**Channel Allocation**

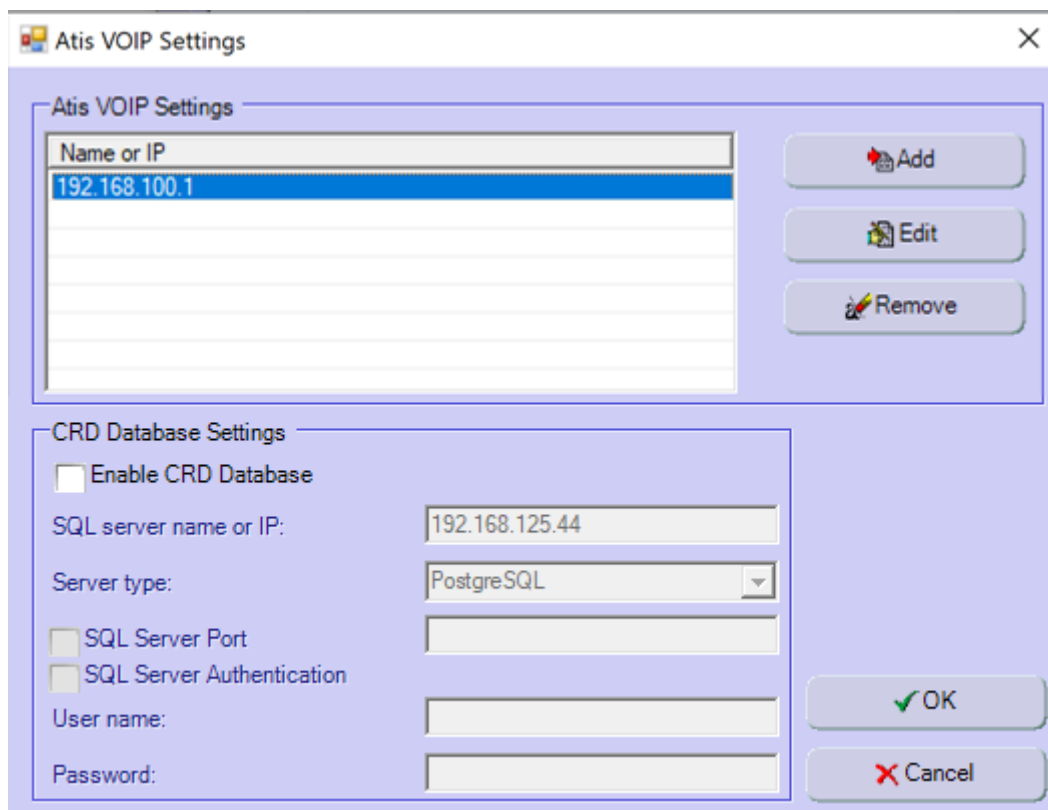
Card ID	Card Name	Channels	I/O Port	Present
0	ATIS_VOIP 192.168.100.1	1 - 30		Yes

☐ Use VOIP & UP0 Interface cards ☒ Use ATIS VOIP ☐ Show full text for channel number ☐ Use RIAB ☐ Use Import Agent

[ATIS VOIP Settings](#) [Restore Order](#) [Apply Changes](#)

Buffer Configuration  
**Channel Allocation**  
General Settings  
Users  
Database Settings  
NAS Settings  
AUX Data Settings  
Restricted OS Access  
R31 Protocol Settings  
ASA-4 Settings  
Save default settings  
Load default settings  
OK  
Apply  
Cancel  
About Configurator

Click on the button **ATIS VOIP Settings**.

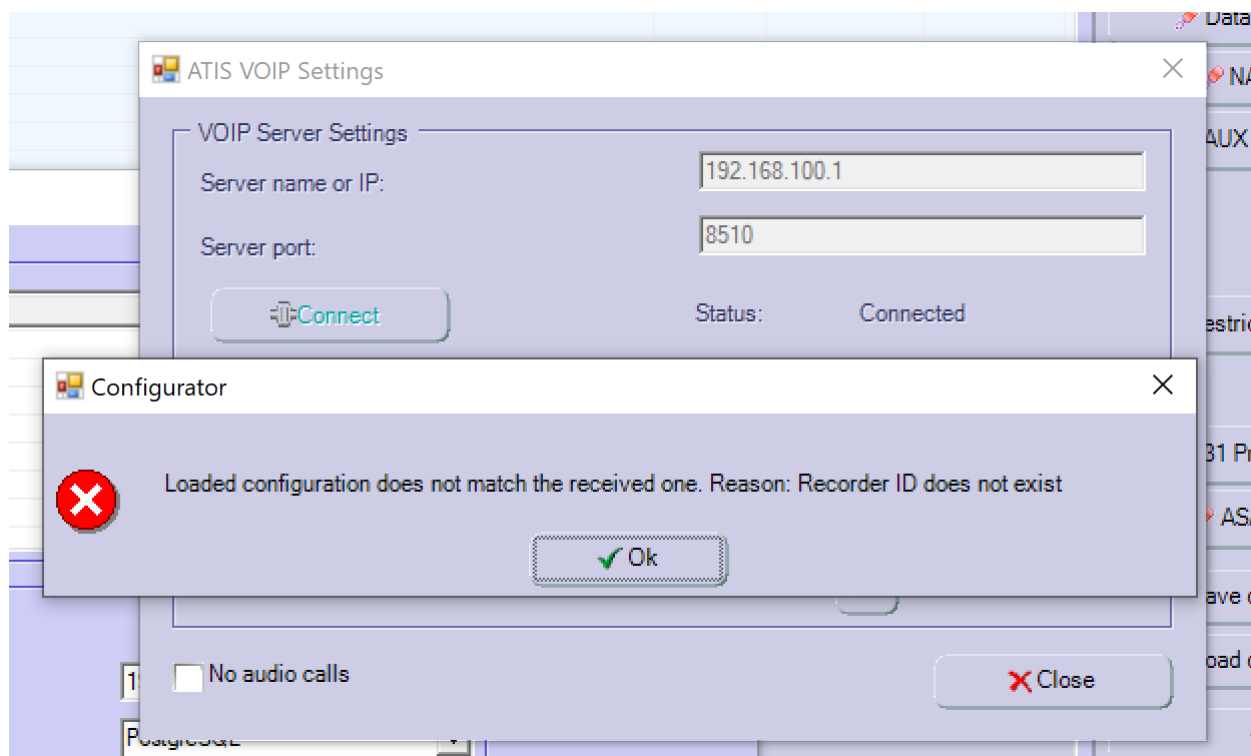


The new dialog displays a list of connections to VDS applications (empty if no VDS has ever been configured). Press the **Add** or **Edit** button

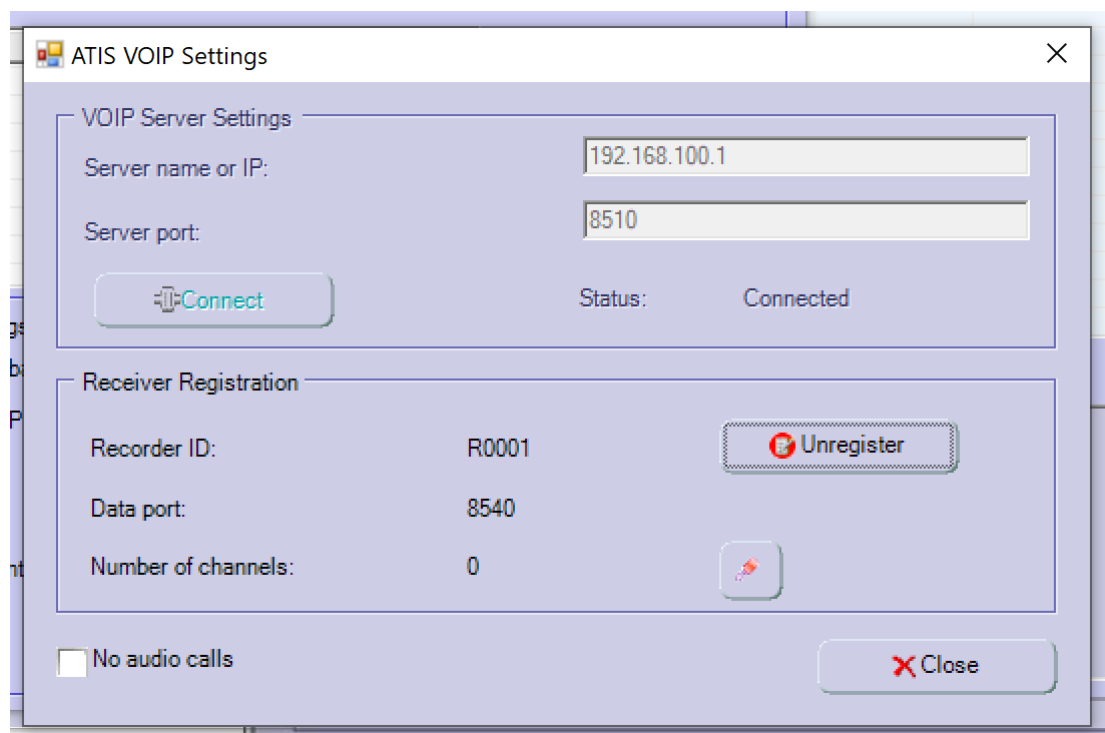
Next dialog displays parameters for the network connection parameters to the VDS.

- In the **Server name or IP** field, enter the IP address of the Interface computer where VDS is installed.
- Then press the **Connect** button

In case of a fresh installation of VDS, may get a warning message informing you that this VDS has never been registered. Just press OK




Once the connection is established, press the Register button:

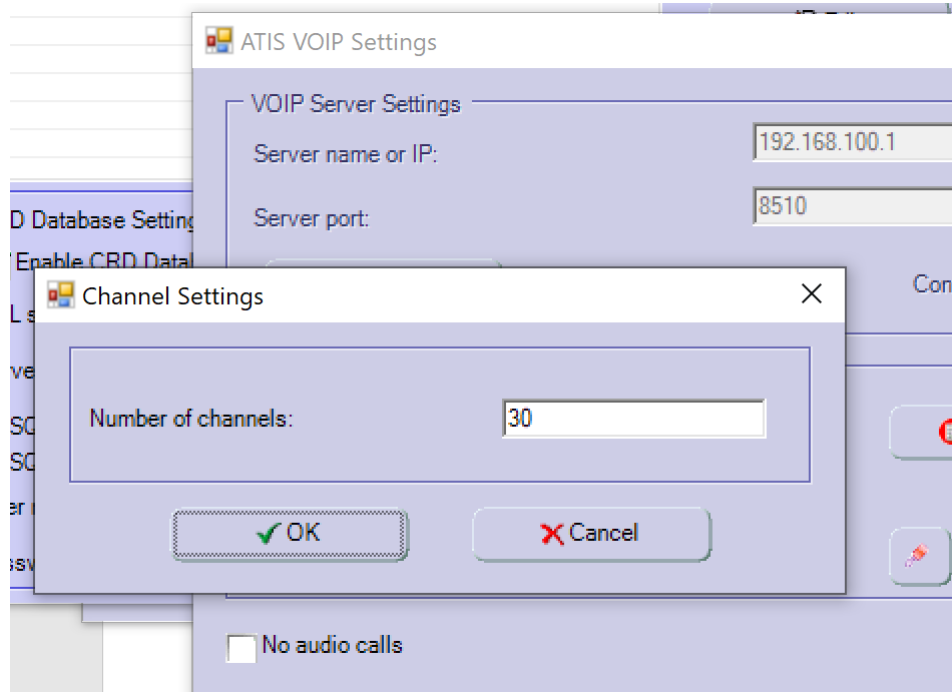


**Note:** The **No audio calls** checkbox allows for keeping records of calls without audio, for example rejected or discarded calls.

Finally, click on the 'Modify channel number' button and enter the number of channels that should be used by the Recorder.

Number of channels:	0	
---------------------	---	---

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



**Note:** The number of channels is object of VoiceCollect licensing. Please, contact VoiceCollect Sales for details.

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

Atis VOIP Settings

Atis VOIP Settings

Name or IP
192.168.100.1

Add Edit Remove

CRD Database Settings

☒ Enable CRD Database

SQL server name or IP: 192.168.100.1

Server type: PostgreSQL

☐ SQL Server Port 5432

☒ SQL Server Authentication

User name: postgres

Password: \*\*\*\*

OK Cancel

VC-MDx Configurator

Channel Allocation

Card ID	Card Name	Channels	I/O Port	Present
0	ATIS_VOIP 192.168.100.1	1 - 30		Yes

Buffer Configuration

Channel Allocation

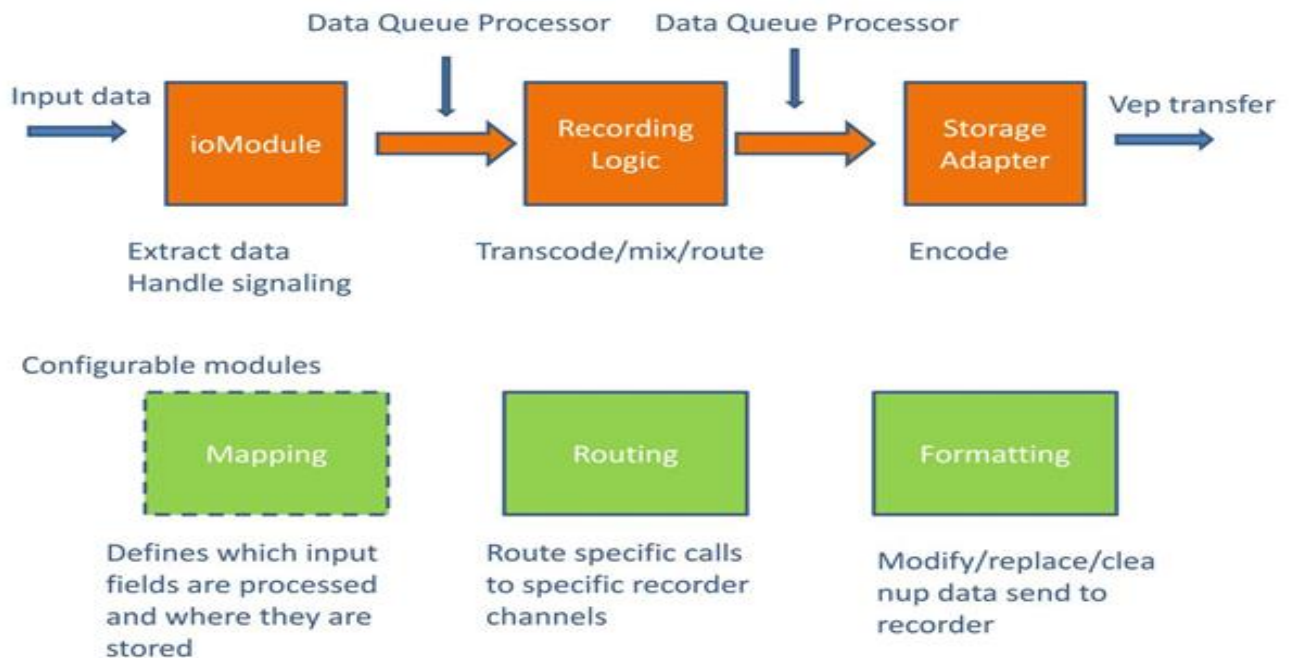
General Settings

Users



## 8 VDS Architecture

Different modules can be configured during the processing of the received data



## 9 Mapping

The Mapping allows the user to change/modify which fields and their priorities are sent to the Recorder.

The default sip passive mapping is configured in the file crd-configuration.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlMappingConfigurationList>
  <!--<XmlMapping incomingFieldName="From" vepFieldName="aParty" /> -->
</XmlMappingConfigurationList>
```

Please refer to the document VDS-II\_Advanced-configuration.pdf

## 10 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. Once the channel is in recording not other call can use it.

The routing configuration is done by editing and configuring the routing-configuration.xml file. The default configuration with 10 channels are shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!--
```

Static routing configuration of VDS input VoIP streams to Recorder channels.  
The filter is taken as a regex value, it can provide powerful matching, it has to be handled with care as syntax is not easy.

The input values used for static routing are aParty and bParty (calling party, called party).  
The filter can contain fully qualified SIP addresses, IP addresses, call numbers or more complex regexps.  
The default generated file with "\*" for all channels is equivalent to a dynamic routing setup.  
Please read ATIS document "VDS-II\_Advanced-configuration\_x.y.pdf" for proper static routing configuration.

#### Filter

\*\*\*\*\*

if filter = "\*" (any): that channel will be added to dynamic channel pool  
if filter = "/" (none): that channel will be added to blocked channel pool  
-->

```
<routingTable>
  <location>ToBeModified</location>
  <recorderList>
    <recorder name="R0001">
      <channelList>
        <channel id="1" filter="*" />
        <channel id="2" filter="*" />
        <channel id="3" filter="*" />
        <channel id="4" filter="*" />
        <channel id="5" filter="*" />
        <channel id="6" filter="*" />
        <channel id="7" filter="*" />
        <channel id="8" filter="*" />
        <channel id="9" filter="*" />
        <channel id="10" 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.

Since release 4.0.3 the routing can be modified without the need of restarting VDS, 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 CallingParty and on CalledParty (the order and the fields can be configured in the spring-config.xml).

Please refer to the document VDS-II\_Advanced-configuration.pdf

## 11 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.pdf](#)

## 12 Troubleshooting

As the VDS is a service or running in a console it is 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.

### 12.1 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 software, but by the Java Runtime Environment they are appearing in the console.

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

The printout should look 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 VoiceCollect GmbH as soon as possible:

- java.lang.NullPointerException
- java.lang.OutOfMemoryError
- java.lang.IndexOutOfBoundsException
- java.lang.ConcurrentModificationException

## 12.2 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 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 starting there is a high chance that something is wrong in the configuration.

#### ClassName :StaticRouter

**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 : XXXXXXXX**

Couldn't find a decoder for the specified codec. Check media description 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 media description 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 (all routing error messages have been moved to info messages)

**Routing messages**

When the recorder registers to the VDS, it gives the VDS 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 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 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 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 didn't received the recorder channels Check channels status on recorder side.

Check VDS status/registration on recorder side.

Verify connection between VDS and recorder.

## **1. Spring-config errors**

The spring-config errors will prevent the VDS 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 is already running and they have a conflicting server address or another program is already using some VDS IP/Port combination.

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

\*\*\* END OF DOCUMENT \*\*\*