VoiceCollect®

Installation Guide

VDS-II
Eurocae Rtsp Active

Version 2007

VoiceCollect GmbH | Justus-von-Liebig-Str. 5 | 61352 Bad Homburg | Germany
voicecollect@voicecollect.de

Table of Contens

1 Document History

Version	Date	Name	Status
1.0	2015-12-15	AF	Initial draft
1.1	2016-03-07	AF	Added simultaneous transmission
1.2	2016-04-22	AF	Java runtime 32 bits version 1.8 minimum
1.3	2020-05-05	VR	Updated until Application version 5.0.6
1.4	2020-06-12	VR	Updated wizard screen Java version 32 bit support
1.5	2020-06-18	YP	Added explanations in 6.3 and 6.5
1.6	2020-07-10	YP	Added routing configuration page screenshot/explanations

2 Purpose

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

3 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	VoIP Decoding System (VoiceCollect)
VEP	VoIP Export Protocol (VoiceCollect)
REGEXP	Regular Expression
VOX	Voice Operated eXchange

4 Introduction

VDS stands for VoIP Decoding System, it is designed to transform an incoming protocol to our standard proprietary recording protocol (Voice 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 Eurocae Rtsp active” supports the ED137 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.…).

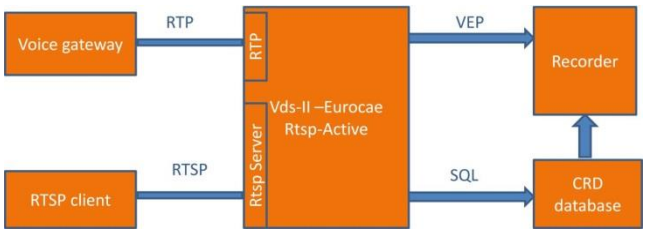
The VDS being a Java application it’s portable on any hardware which can run a java runtime machine, but it has some limitation due to some codecs which are only available for windows (dll) for now (tetra,g726,g723).

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-Eurocae-Rtsp-Active



The VDS:

- Handles all RTSP messages (recover CRD, Start/Stop recording)
- Receives Audio through RTP listener

6 Installation

6.1 Installation of VDS

The VDS package is provided as a Windows installer named **vds-eurocae-rtsp-active-x.x.x_setup.exe** to be installed on VoiceCollect Interface Computer. The screenshots below explain the most important steps.

NOTE:
if you want to use the CRD database with the VDS, the SQL version must be minimum 10 (SQL 2008) if using Microsoft SQL database.

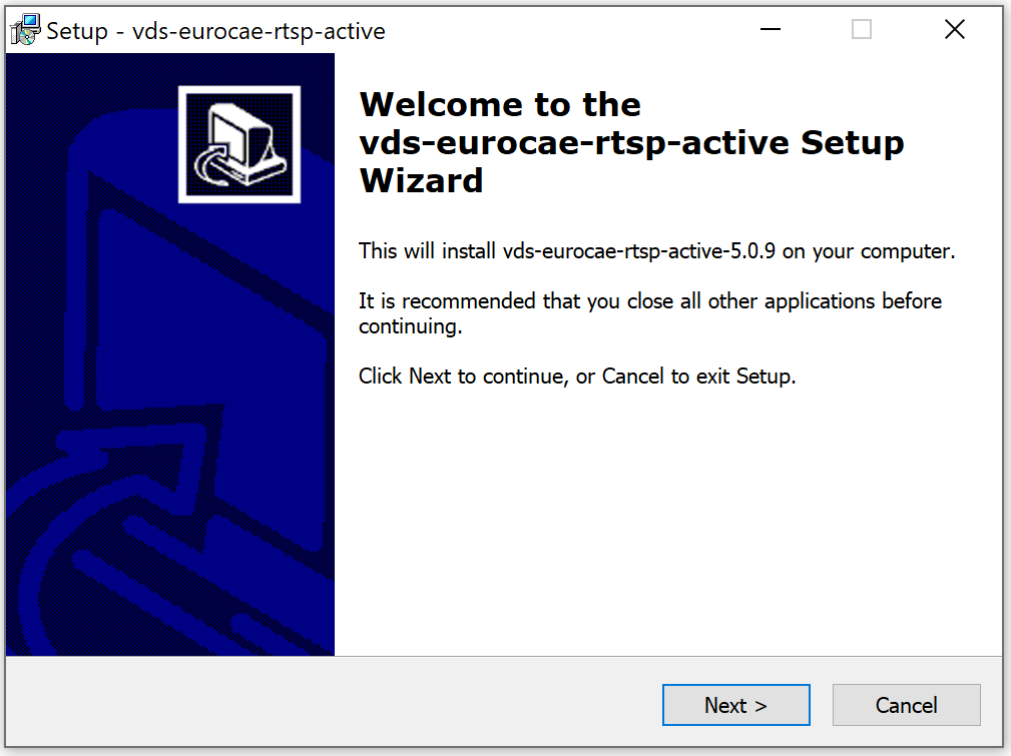
NOTE:
A 32-bit/64-bit version of the Java Runtime Environment with java 1.8 or above must already be installed.

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

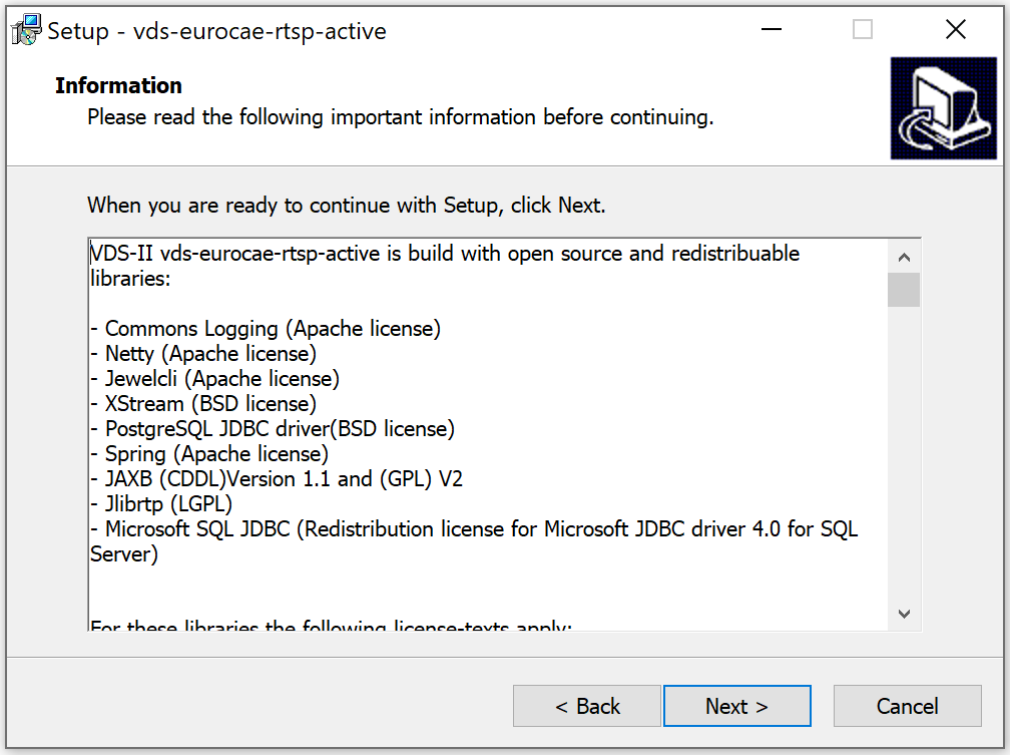
Locate the file **vds-eurocae-rtsp-active -x.x.x_setup.exe** and double-click on the icon:

vds-eurocae-rtsp-active-5.0.9_setup.exe

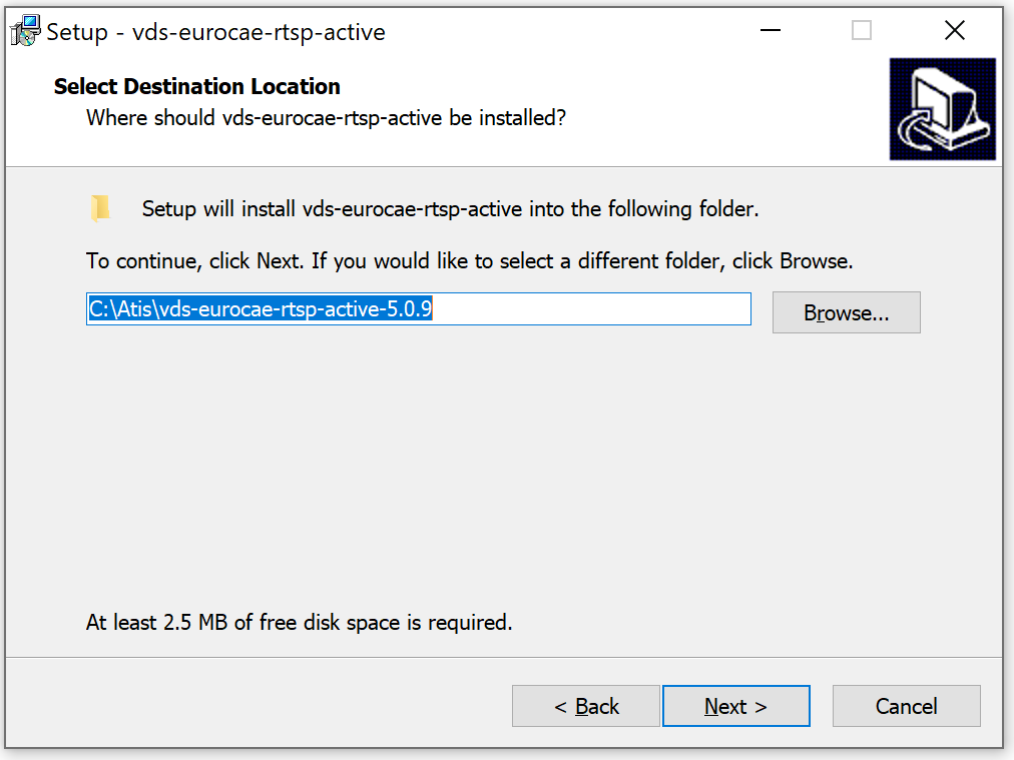
The following dialog box appears, press next:



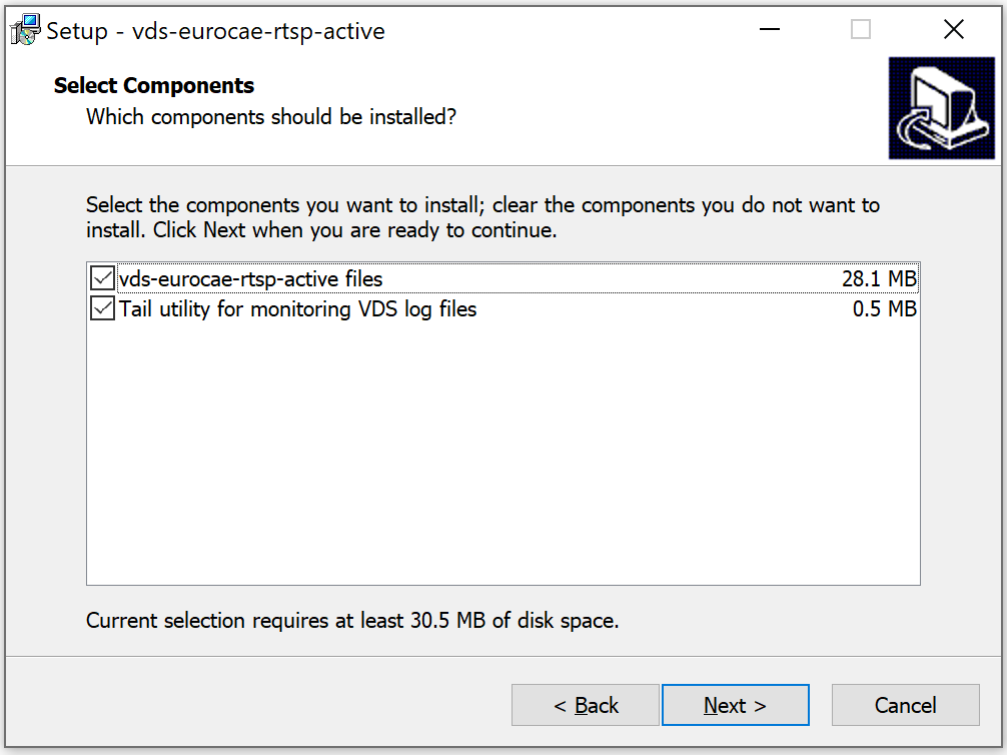
Select the Next button to proceed the installation:



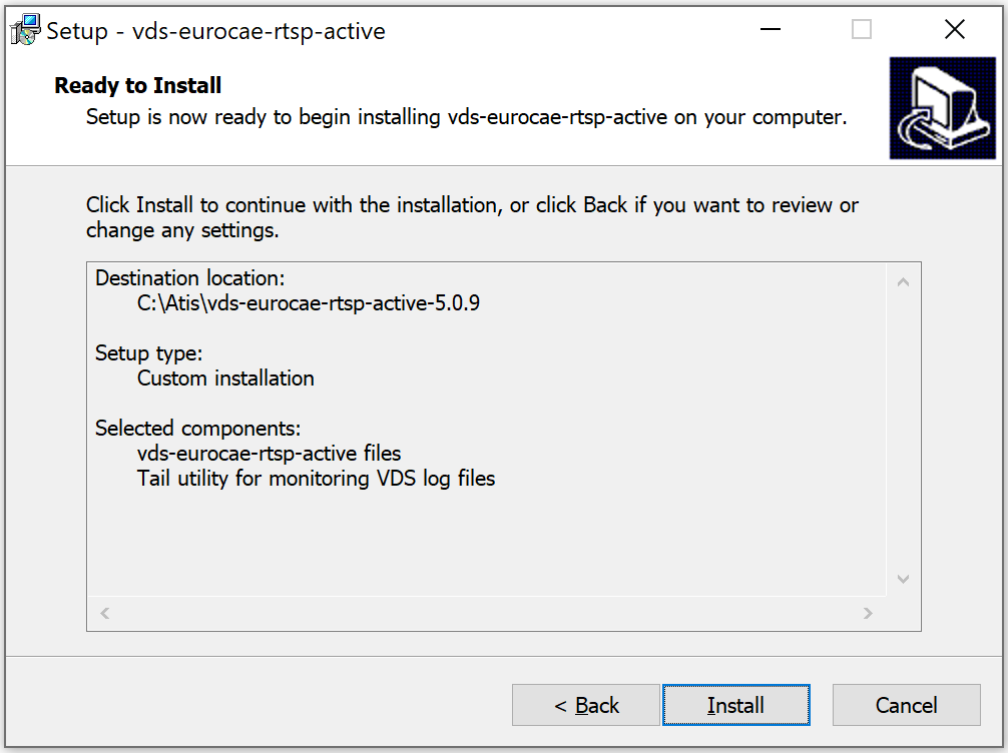
Select the directory where VDS-II will be installed in:



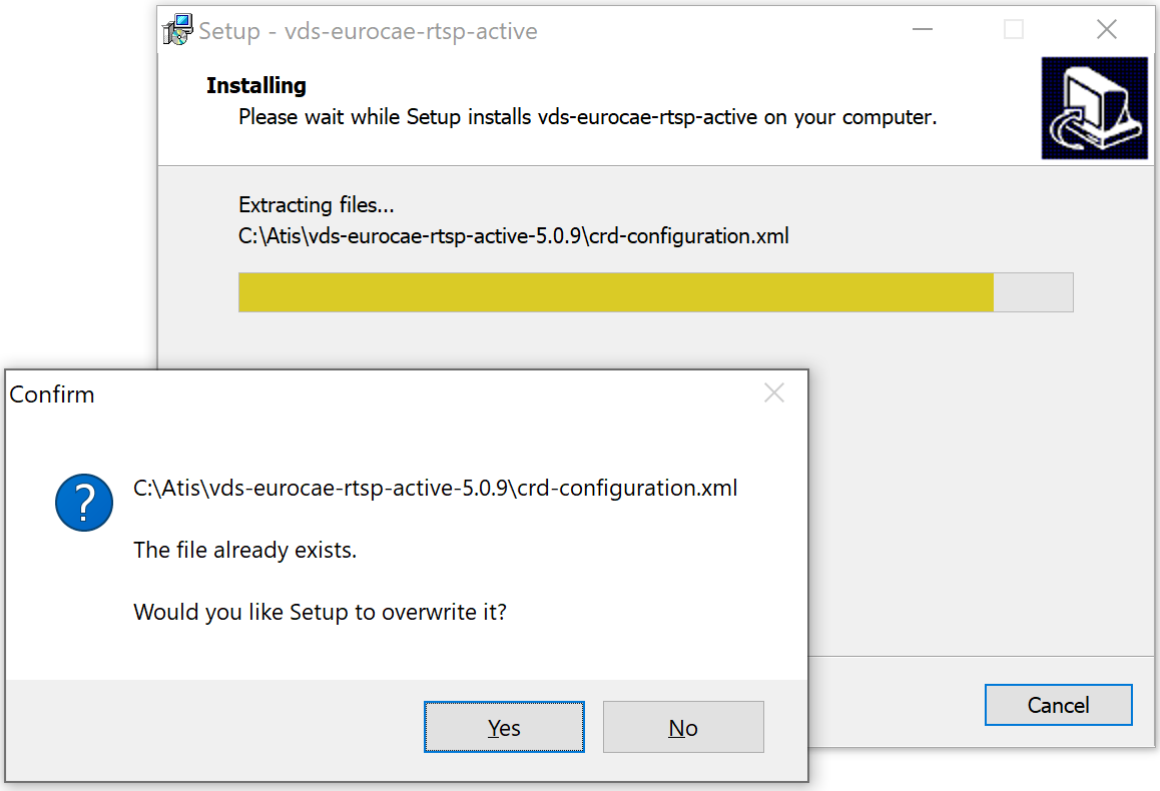
Select the components that will be installed:

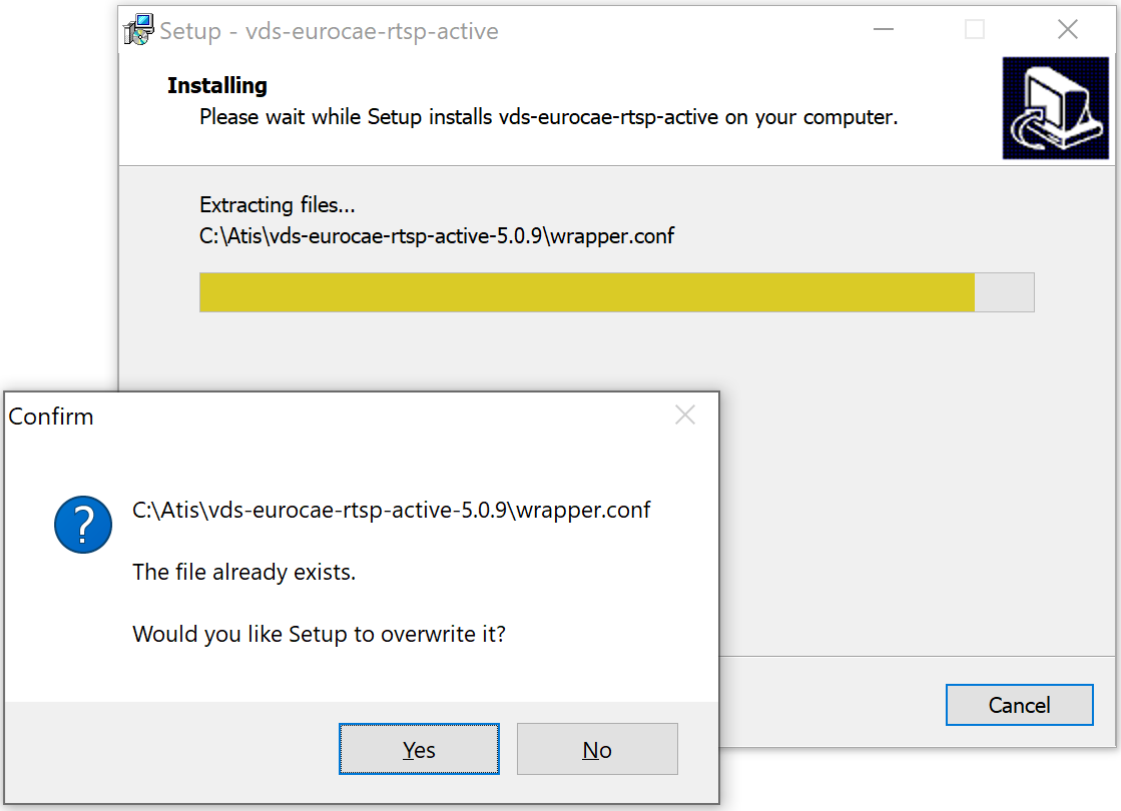


Click on Install after reviewing the configured settings.

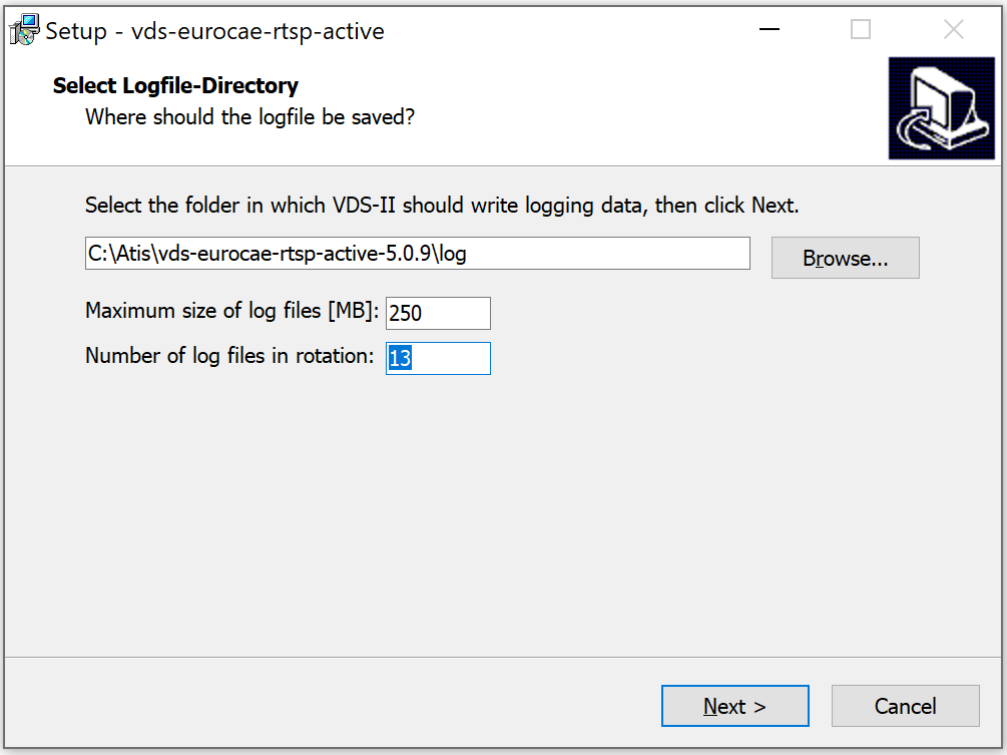


Note: If you are performing an update (installation in the same directory as the one of an existing VDS instance) rather than a fresh installation, you will be asked to choose whether you want to keep your configuration files or if you want to create new default ones. In most cases, you will not choose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

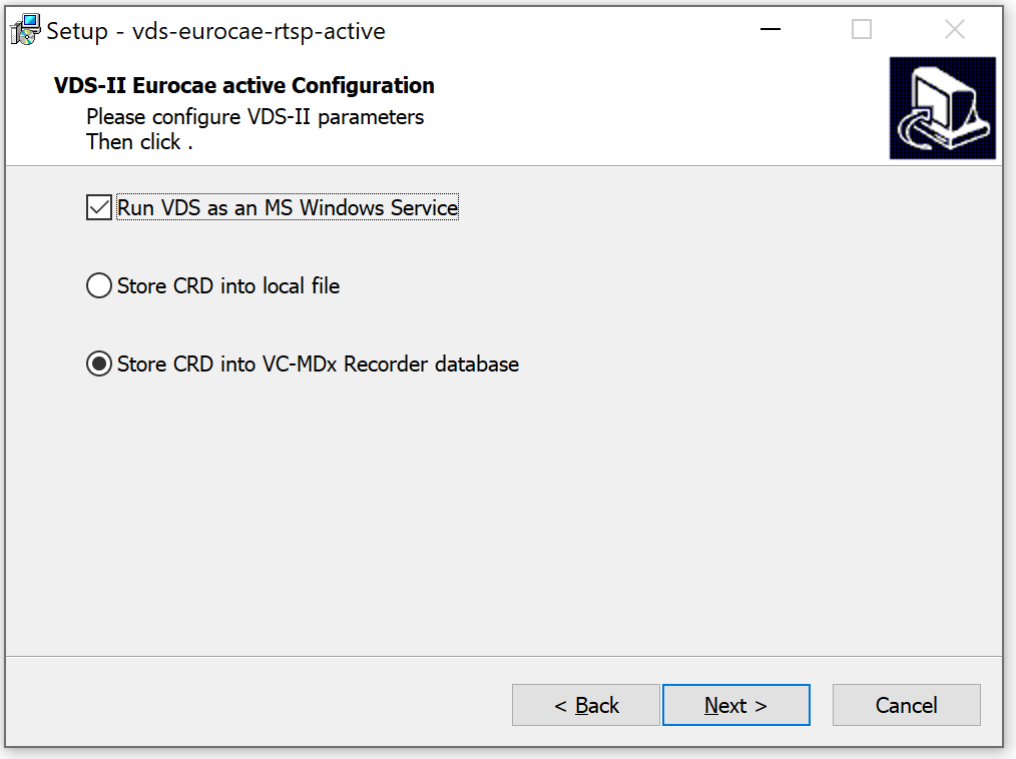

Setup - vds-eurocae-rtsp-active
Installing
Please wait while Setup installs vds-eurocae-rtsp-active on your computer.
Extracting files...
C:\Atis\vds-eurocae-rtsp-active-5.0.9\crd-configuration.xml
Cancel
Confirm
C:\Atis\vds-eurocae-rtsp-active-5.0.9\crd-configuration.xml
The file already exists.
Would you like Setup to overwrite it?
Yes
No


Setup - vds-eurocae-rtsp-active
Installing
Please wait while Setup installs vds-eurocae-rtsp-active on your computer.
Extracting files...
C:\Atis\vds-eurocae-rtsp-active-5.0.9\wrapper.conf
Cancel
Confirm
C:\Atis\vds-eurocae-rtsp-active-5.0.9\wrapper.conf
The file already exists.
Would you like Setup to overwrite it?
Yes
No

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



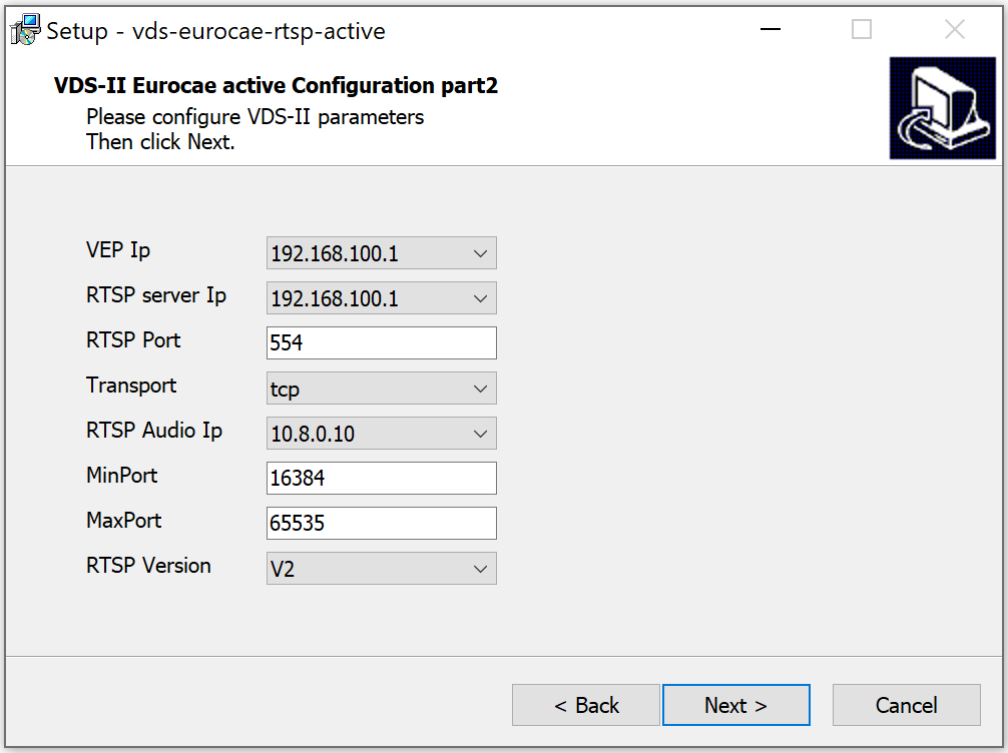
6.3 VDS –Eurocae-Rtsp-Active configuration parameter



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).

Store Crd into local file(radio button)
This setting is not recommended as all crd data will be locally stored into a txt file. Useful for debugging and testing. **Refer Section 12.9**

Store Crd into VC-MDx Recorder database(radio button)
This is the default. All crd data will be stored into the VC-MDx recorder database.



Parameters:

- **Vep Ip:** IP Address of the network interface connected to VC-MDx.
- **RTSP Server IP**: IP Address for incoming RTSP connection from the VCS.
- **RTSP Port**: Listening port for RTSP connections.
- **Transport**: Selection of the transport protocol for RTSP connection (TCP or UDP)
- **RTSP audio IP**: IP address for incoming RTP audio data.
- **RTP min port / max port** : Allowed port range for incoming RTP connections
- **RTSP version**: Selection of the version of RTSP protocol (V1 or V2).

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 'theIoModule' bean definition.

Note: both IPv4 and IPv6 addresses can be selected.

Note on RTSP Version:
VDS complies to RTSP V1.0 (RFC2326) and to RTSP V2.0 (RFC7826). In case of doubt, we recommend to select V1.

After the installation, to switch from one version to another, edit the file **spring-config.xml** file and change the value of the parameter **rtspVersion** according to the desired value (V1 or V2). VDS needs to be restarted afterwards for the changes to take effect.

This parameter essentially controls the transport header parameters that are set in replies to SETUP requests:

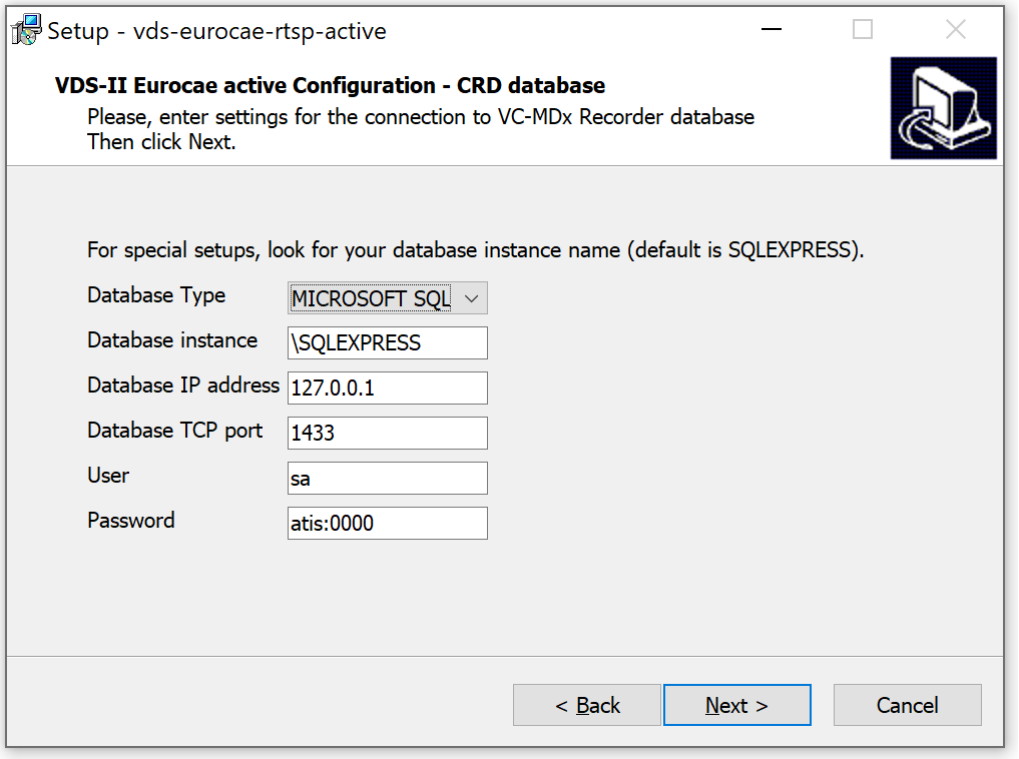
With RTSP V1, the parameters **destination** and **server_port** are used to specify the listening IP address and port for the delivery of RTP streams.

With RTSP V2, the parameter **src_addr="ip:port"** is used instead.

6.4 CRD database-configuration

VDS can store CRD records in a MS SQL or Postgres database. The purpose of this screen is to configurae the connection to the database.

Note: The VC MDx recorder database setting should match this CRD Database configuration



Database Type: choose the database type
- MICROSOFT SQL (default)
- POSTGRES 9.x

Database instance: This name defines which instance of the database to be used. Change it with care. Default SQLEXPRESS (for Microsoft SQL type)

Database IP address: IP of the database server instance (usually same as the one of the recorder)

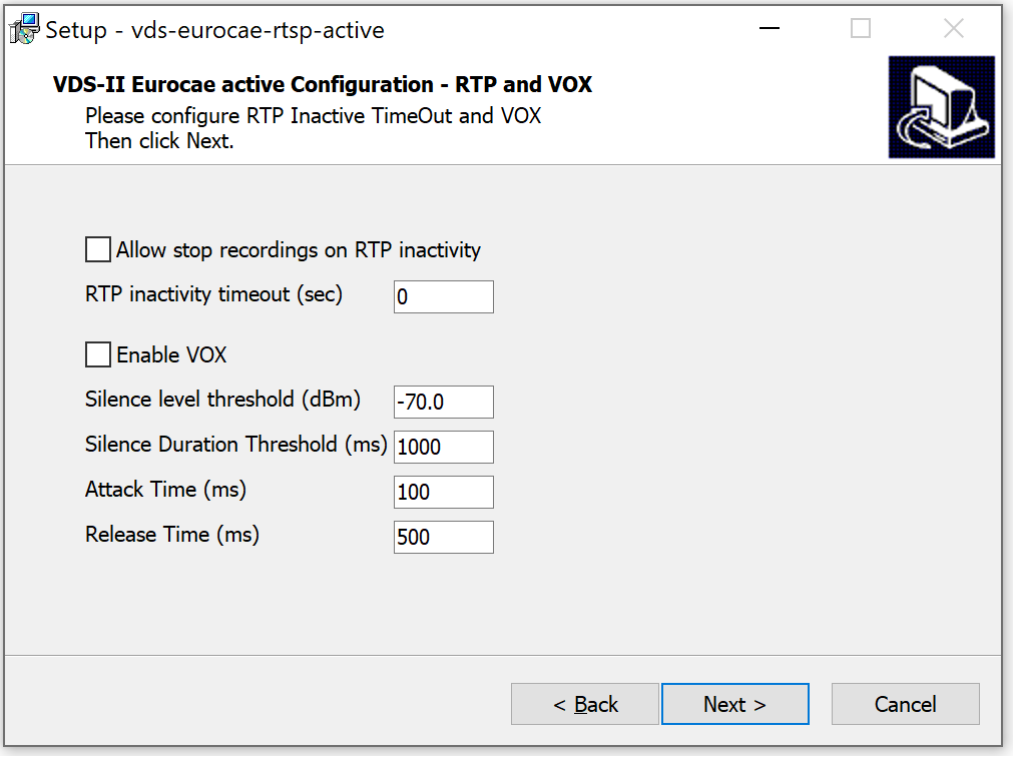
Database TCP Port: Default is 1433

User: Default user for crd database

Password: Default password for crd database

6.5 RTP Inactive Timeout and VOX Configuration

This section allow configuration of recording start and stop triggers.



By Default the RTP Inactivity and VOX configuration will be disabled. It can be enabled and the relevant configurations can be done in this interface.

- **RTP Inactivity Timout(Sec)**: Based on this configuration the Inactive RTP Sessions will be closed and recording will be stopped. This setting can be useful to close recordings for stations that have been physically disconnected during a communication.
- **Vox settings**:

Some VCS (eg: Frequentis) send a RECORD request and continuous flow of audio data for all configured channels. And it is the responsibility of the recorder to start and stop the recording based on detection of voice in the audio streams. When such a recording mode is required, the VOX option shall be activated.

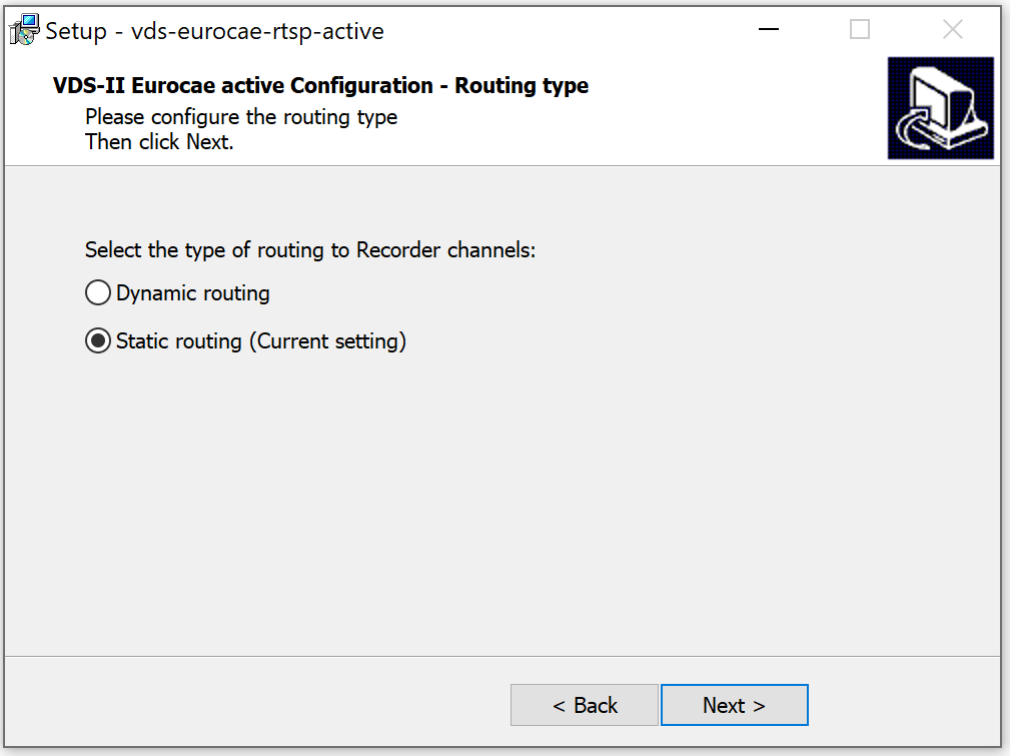
  - **Silent Level (dBm):** Recording will start when the power measured in audio samples will be higher than this threshold. Setting this value too loo may result in recording silence. Setting it too high may result in loosing some low volume audio data.
  - **Attack time (ms):** When audio signal is detected, the recording will contain audio data that was captured during the attack time ms prior to signal detection.
  - **Silence Duration Threshold (ms):** When the audio signal goes below the detection threshold, VDS will wait for this amount of time prior to stopping the recording. This parameter avoid creating many recording when short silences occur.

- **Release Time (ms):** When the conditions are met to stop a recording, VDS will continue to record audio data for this amount of time prior to stop a recording. This is to avoid abrupt end of audio signal.

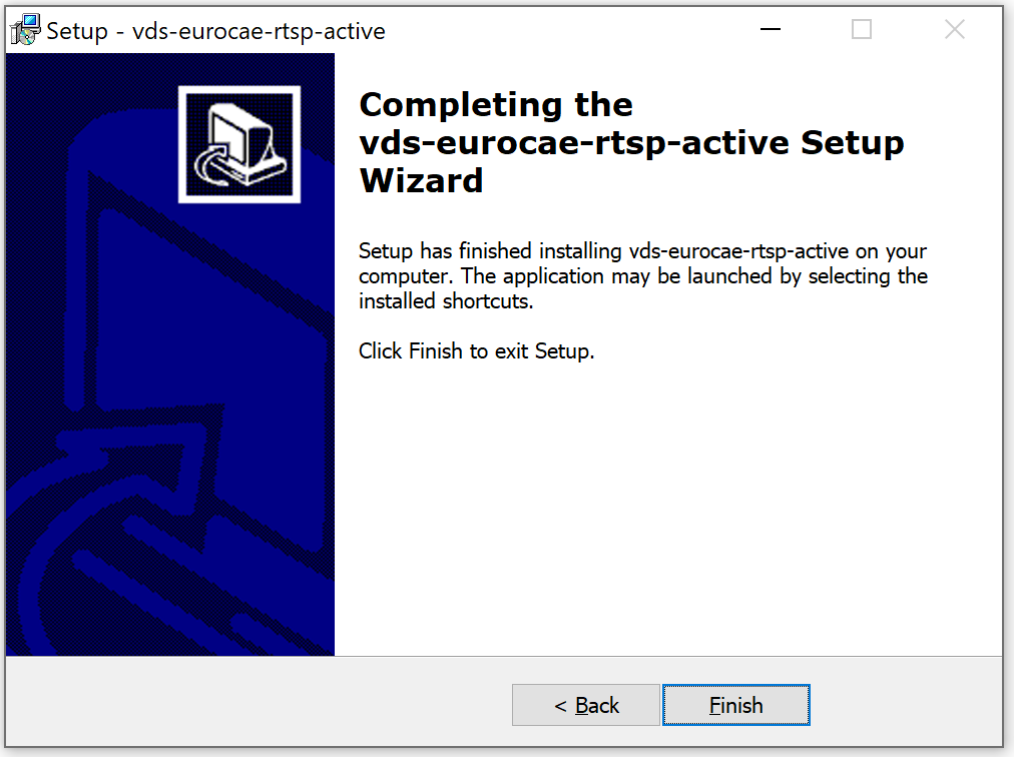
6.6 Routing type configuration

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.



Click on Finish the Setup to close the Installation Wizard.

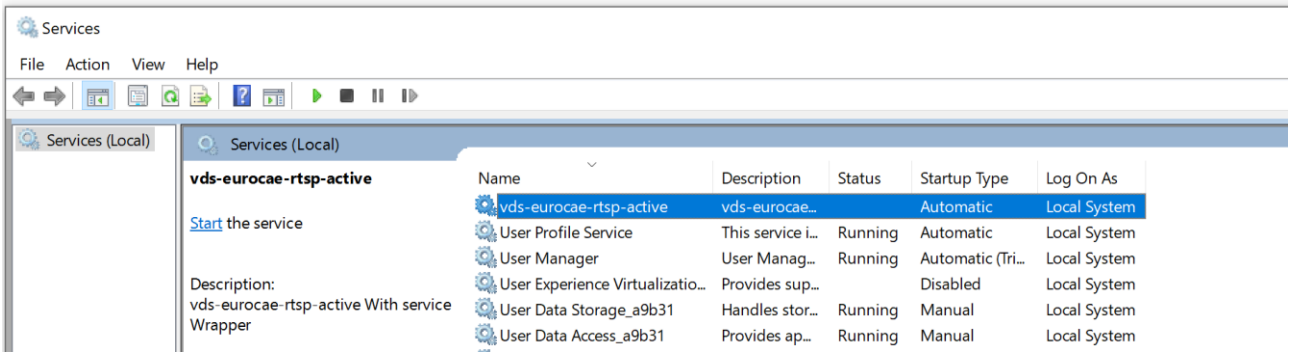


6.7 Verifications

6.7.1 Checking the Service status

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

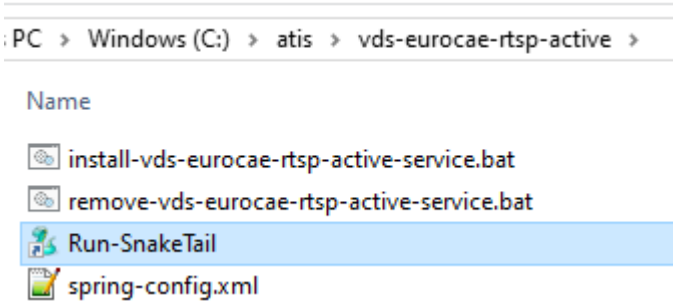
Note: This should be done only when 'Run VDS as an MS Windows Service option' is checked during the installation.



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

6.7.2 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.



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

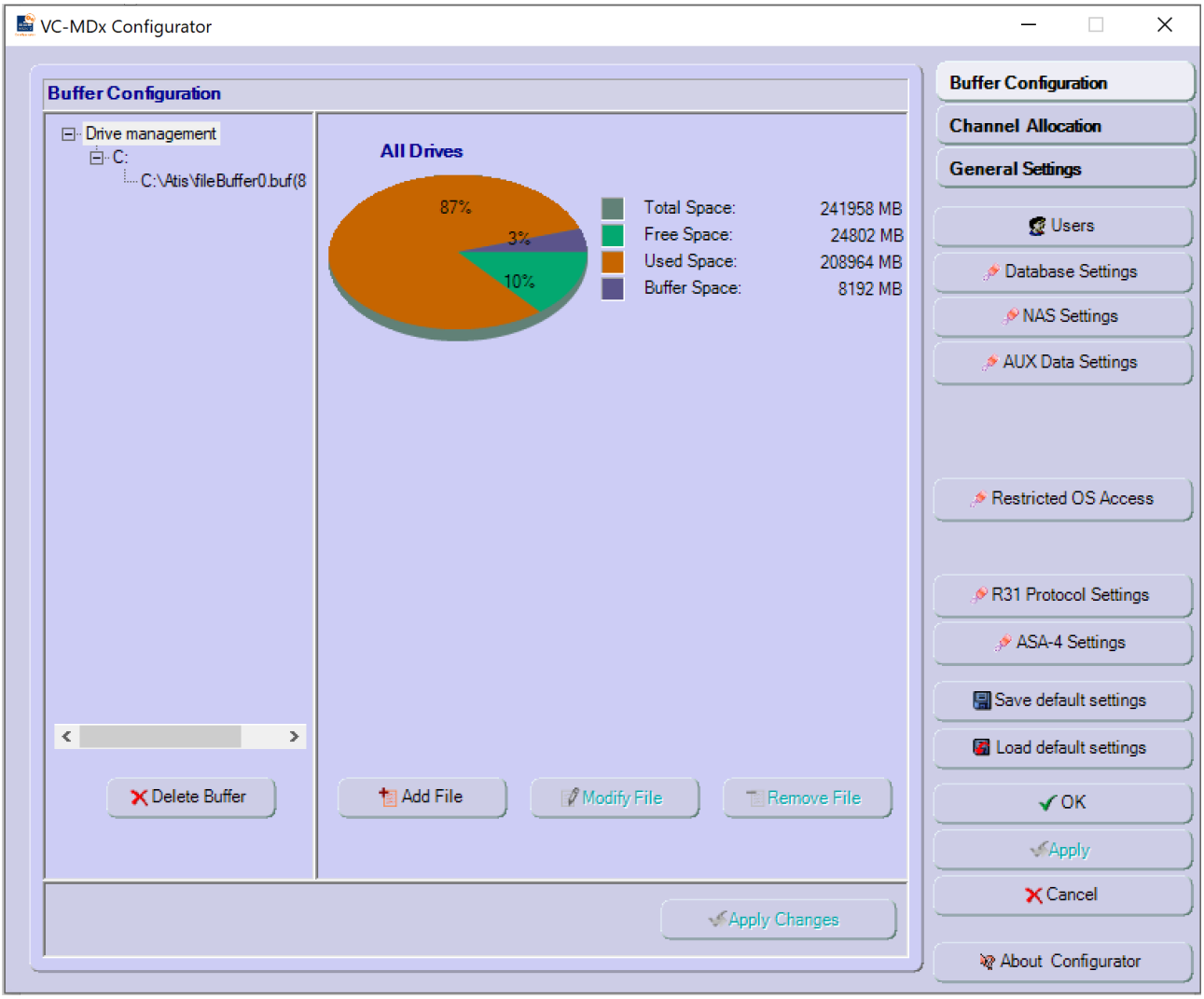
7 Configuration of VC-MDx Recorder

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.

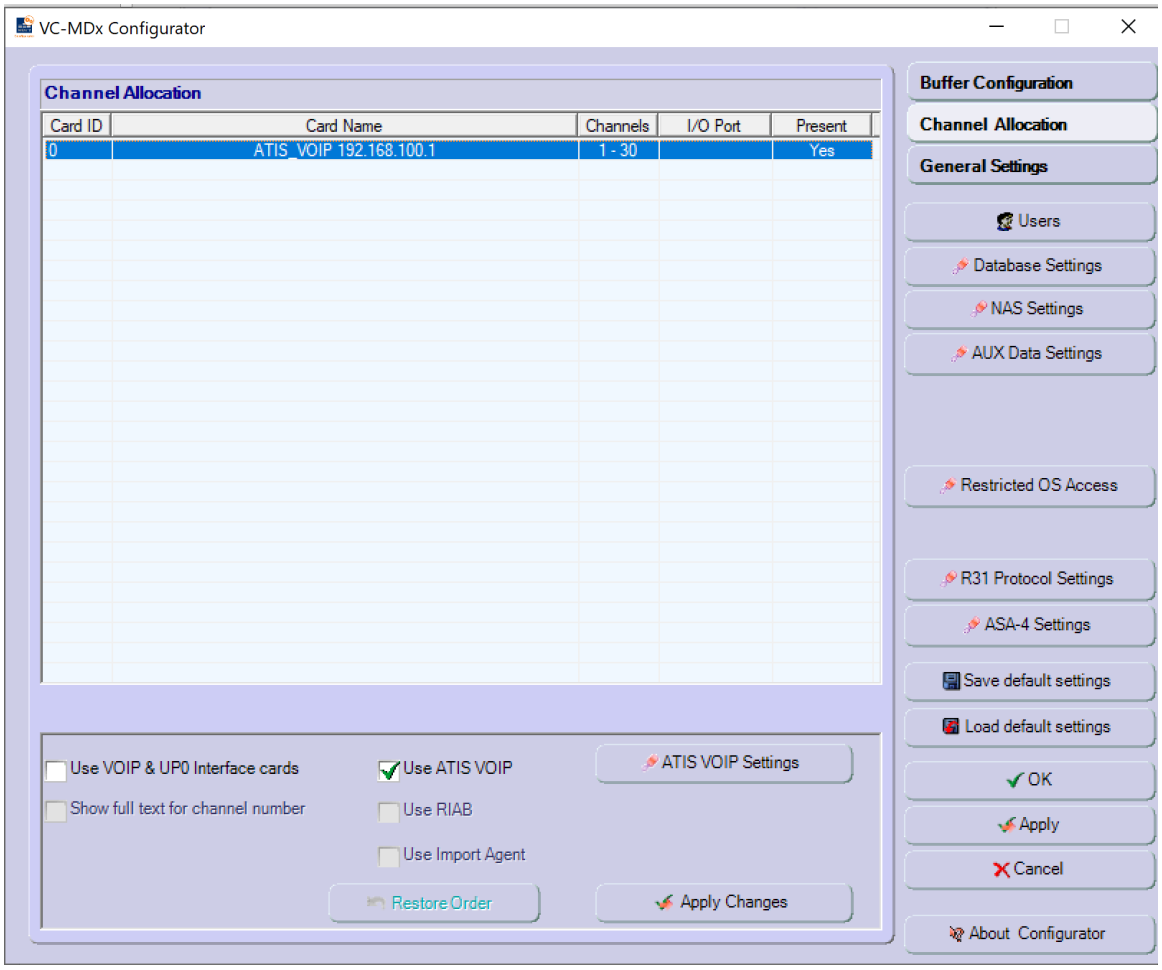
Configurator.exe

After authentication, the following dialog box appears:

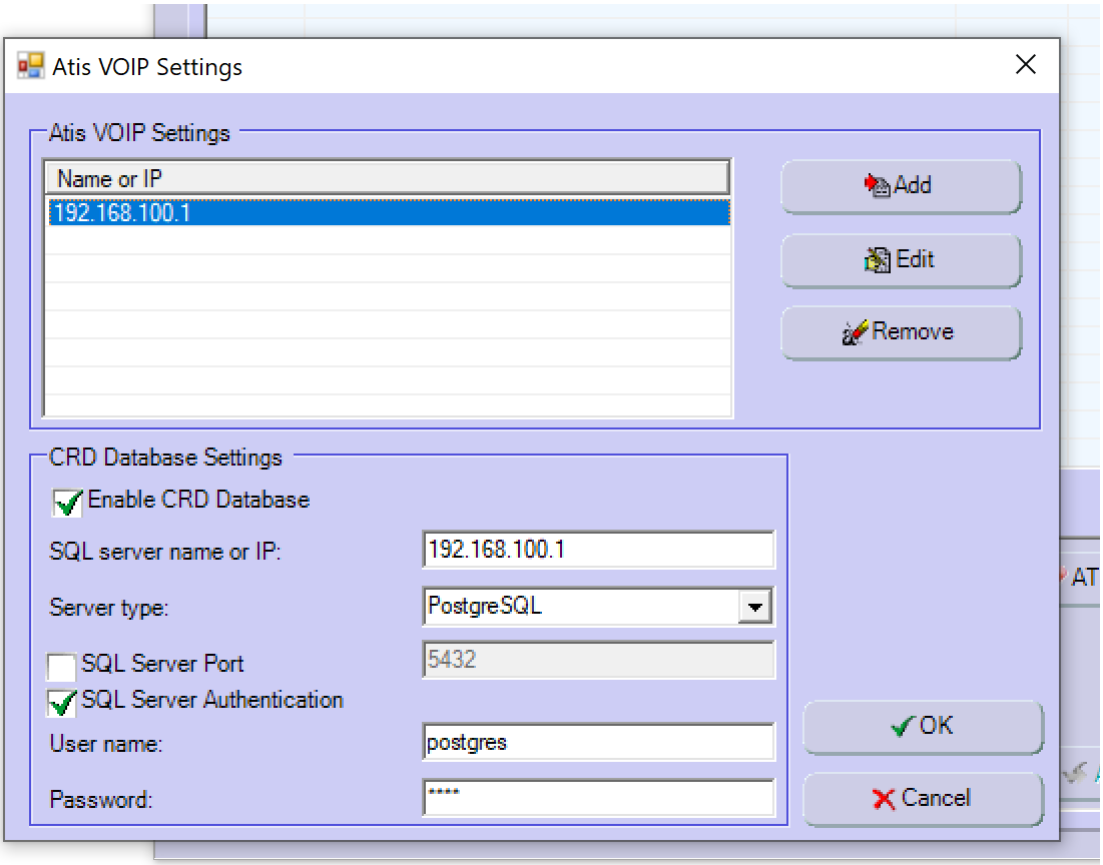


7.1 Creation/Edition of ATIS VOIP Server

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



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

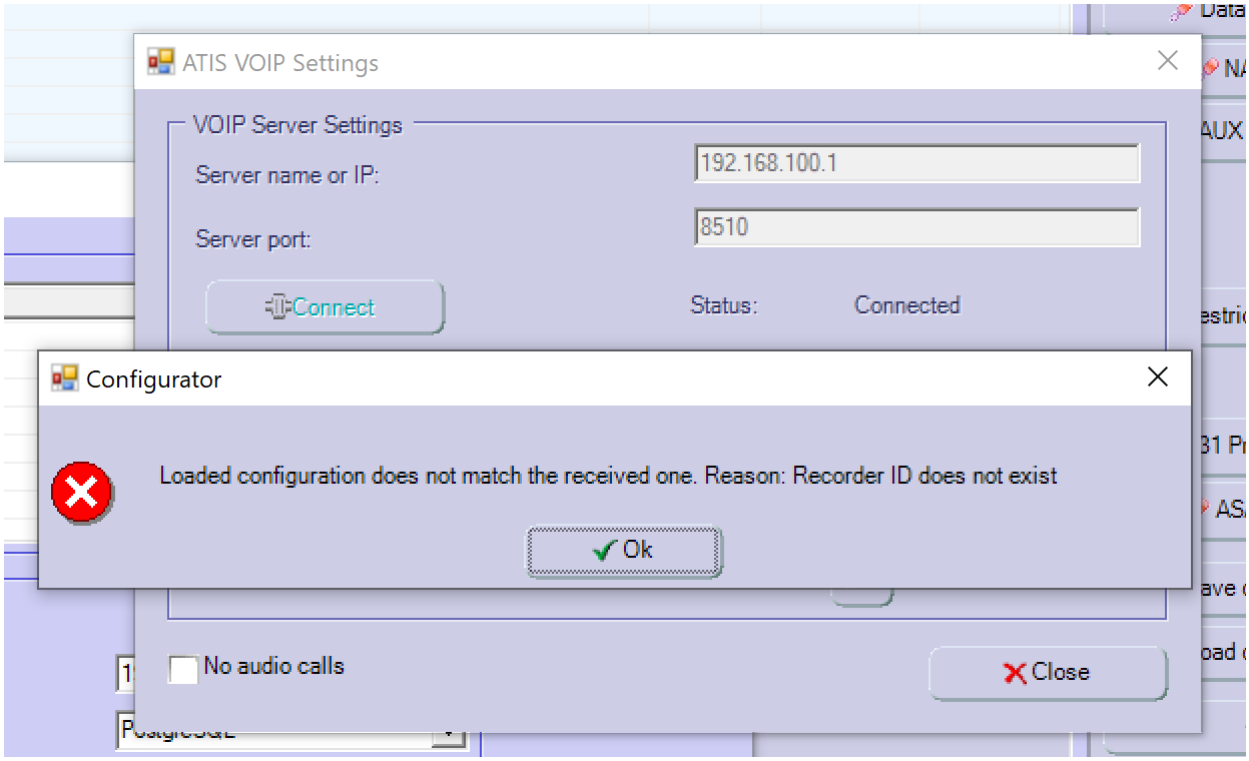


The new dialog displays a list of connections to VDS applications (empty if no VDS has ever been configured). Press the **Add** or **Edit** button

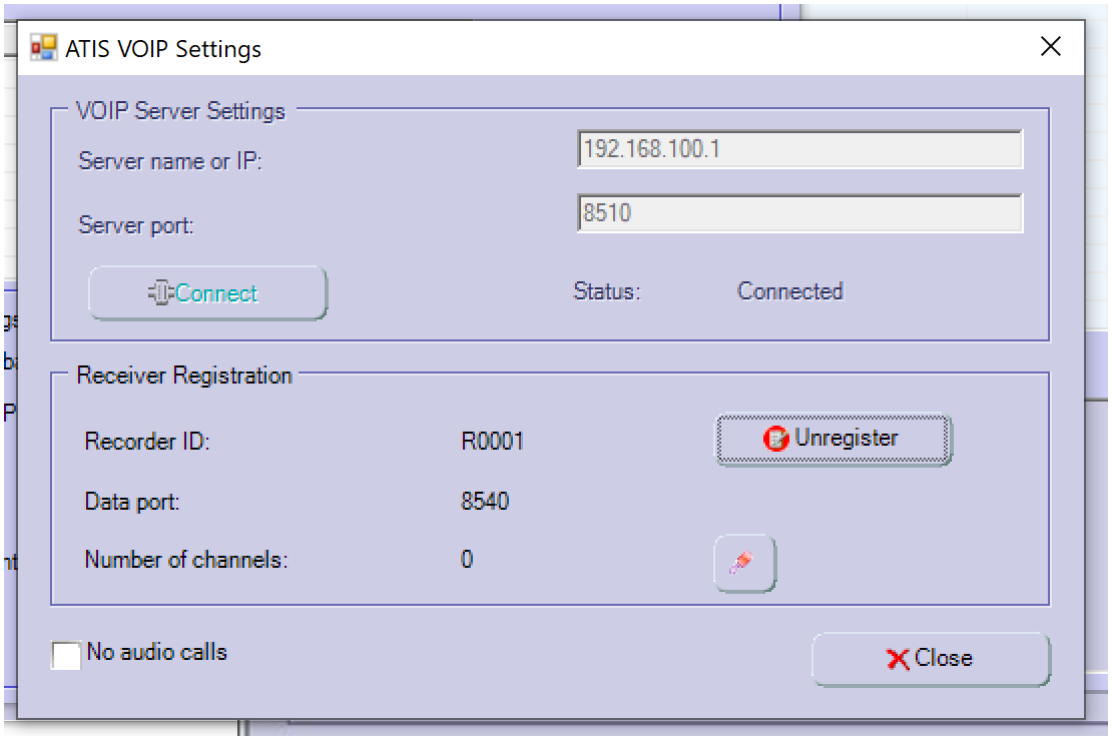
Next dialog displays parameters for the network connection parameters to the VDS.

- 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

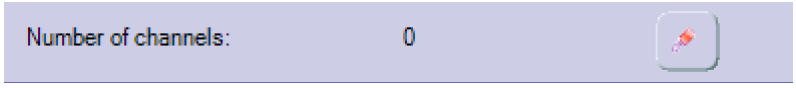


Once the connection is established, press the Register button:

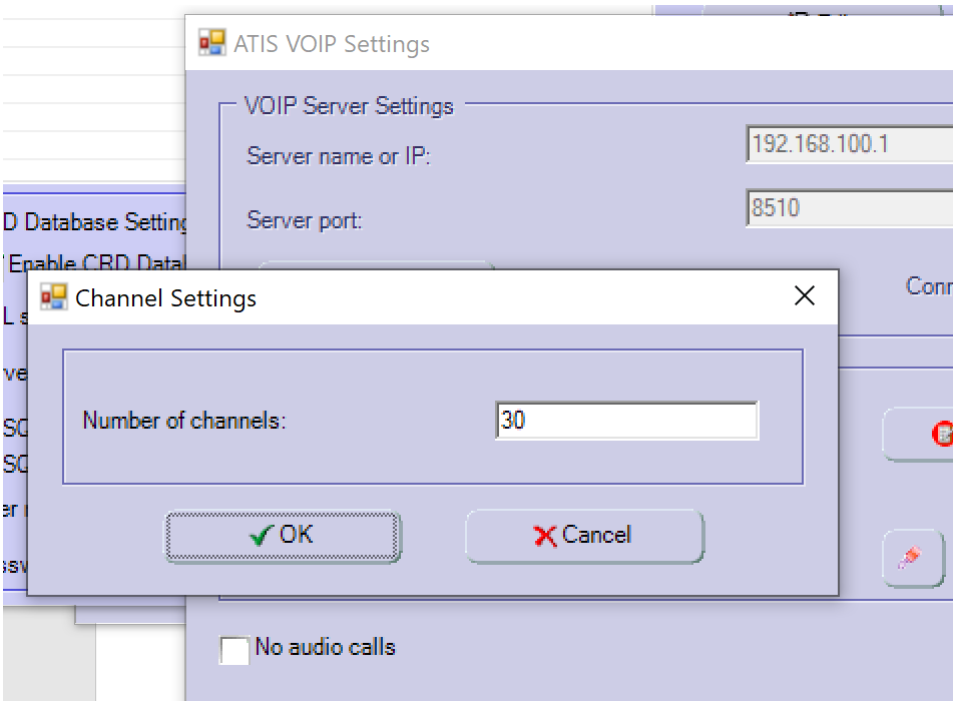


***Note**: The **No audio calls** checkbox allows f or keeping records of calls without audio, for example rejected or discarded calls.*

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

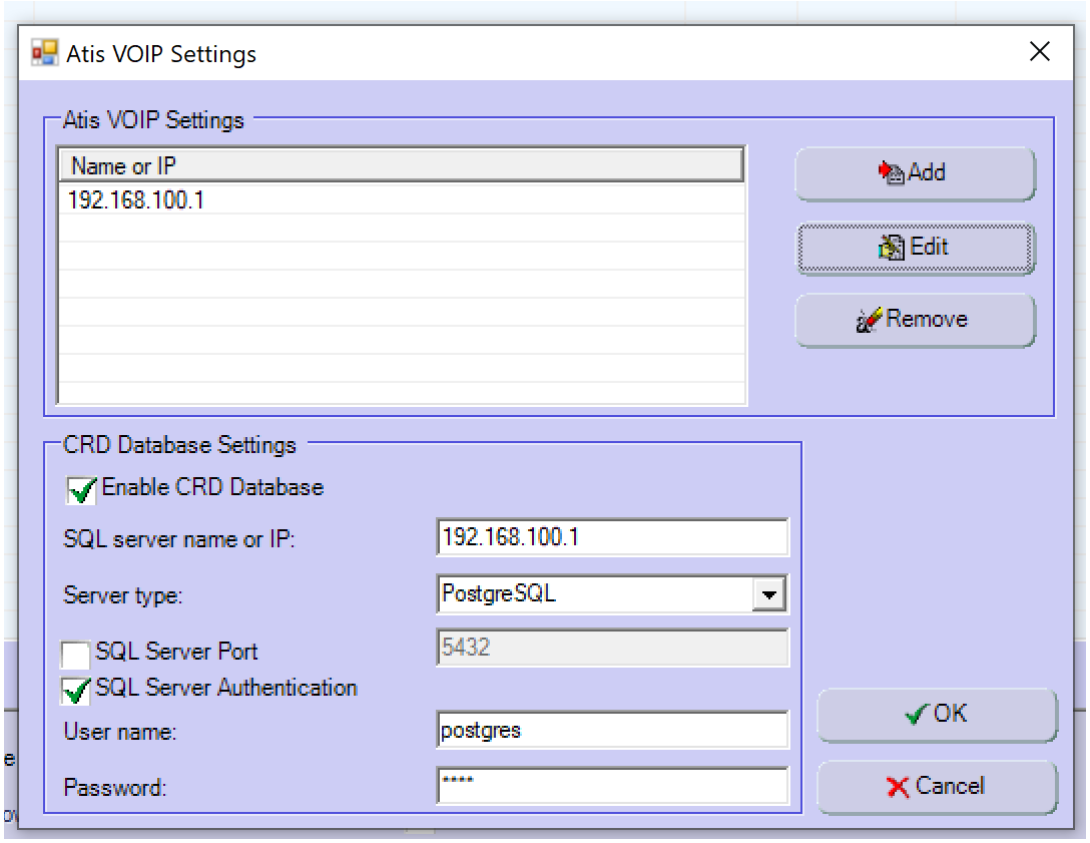


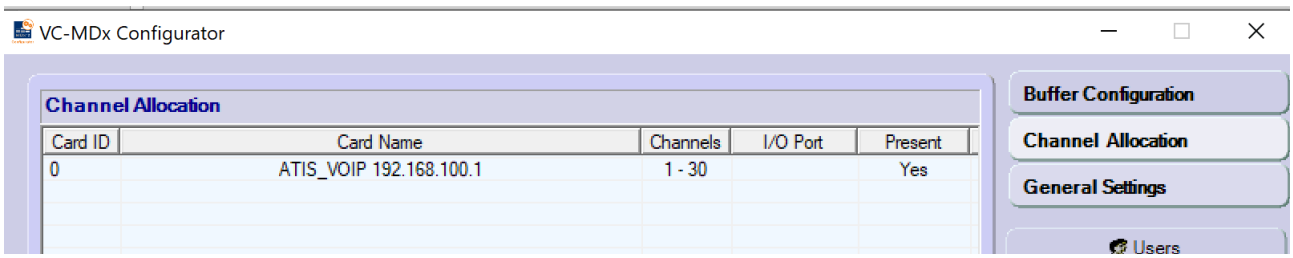
Following dialog appears where you can enter the number of channels:



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:





7.2 CRD Database

Only VDS which are collecting enhanced/additional CRD (call related data) are using the CRD database (like vds-eurocae-rtsp-active)

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

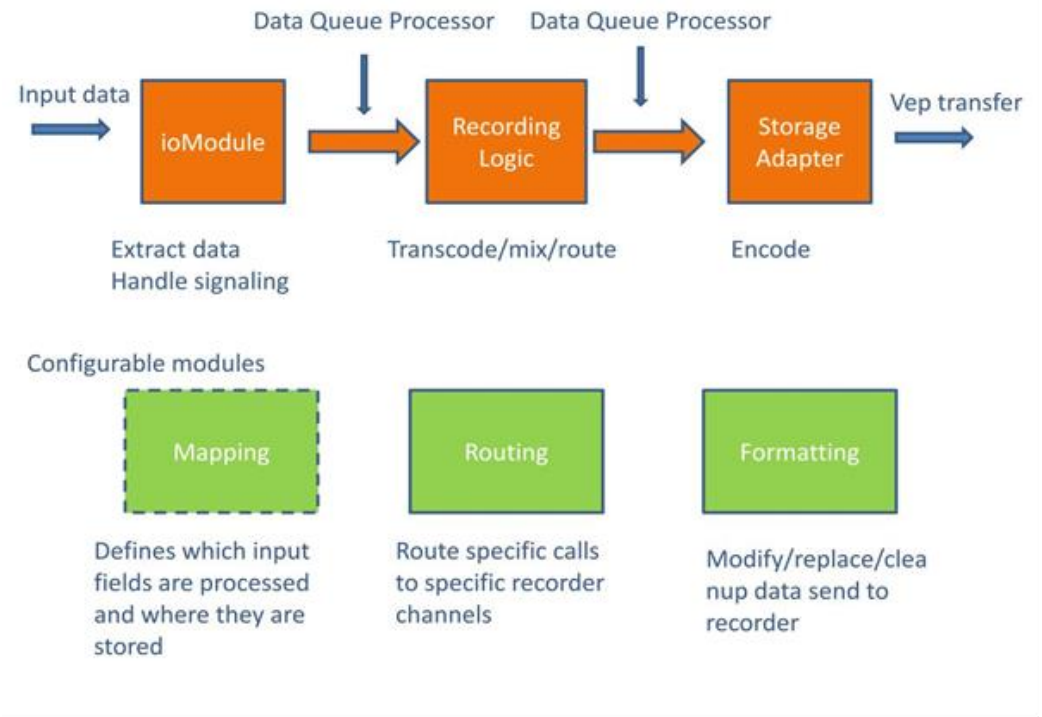
Usually, this database is in the Recorder computer along with main record database, therefore the IP address of the Recorder can be entered in the field named **SQL server name or IP**.

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.

8 Configuration

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 database

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

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<XmlMappingConfigurationList>
    <XmlMapping incomingFieldName="ConnectedNr,CalledNr"        vepFieldName="bparty"     />
    <XmlMapping incomingFieldName="FrequencyID,CallingNr"       vepFieldName="aparty"     />
    <XmlMapping incomingFieldName="ClientId"                    vepFieldName="UserId"     />
    <XmlMapping incomingFieldName="CallRef"                     vepFieldName="dtmf"       />
    <!-- RemoteAddress is not a ed137 field but coming directly from rtp connection-->
    <XmlMapping incomingFieldName="RemoteAddress" vepFieldName="routingIpAddr" /> <!—
    SimultaneousTransmission is extracted from ED137 operations -->
    <XmlMapping incomingFieldName="SimultaneousTransmission" vepFieldName="shortComment"/>
</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

```
<?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.

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 Installed Files

12.1 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.

12.2 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).

12.3 Runtime.xml

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

12.4 Batch files

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

- install-vds-eurocae-rtsp-active-service.bat Install the service
- remove-vds-eurocae-rtsp-active-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.

12.5 Dll Files

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

12.6 Wrapper files

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

12.7 Jar file

This is the main VDS software

12.8 Log files

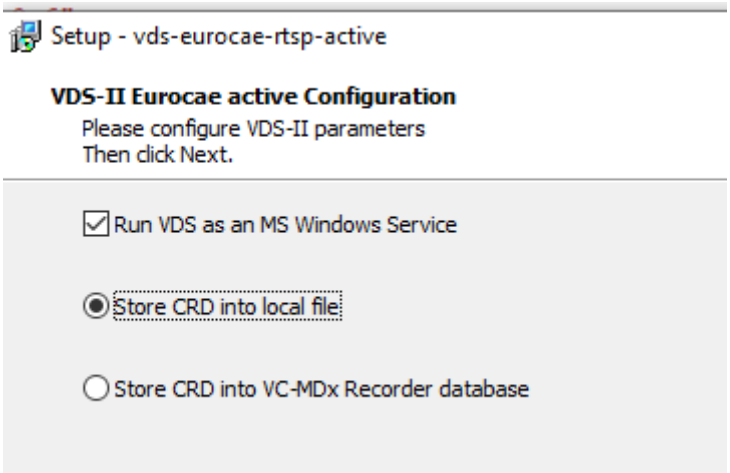
Two log files are created into the directory chosen during installation

- vds-eurocae-rtsp-active.log VDS log file

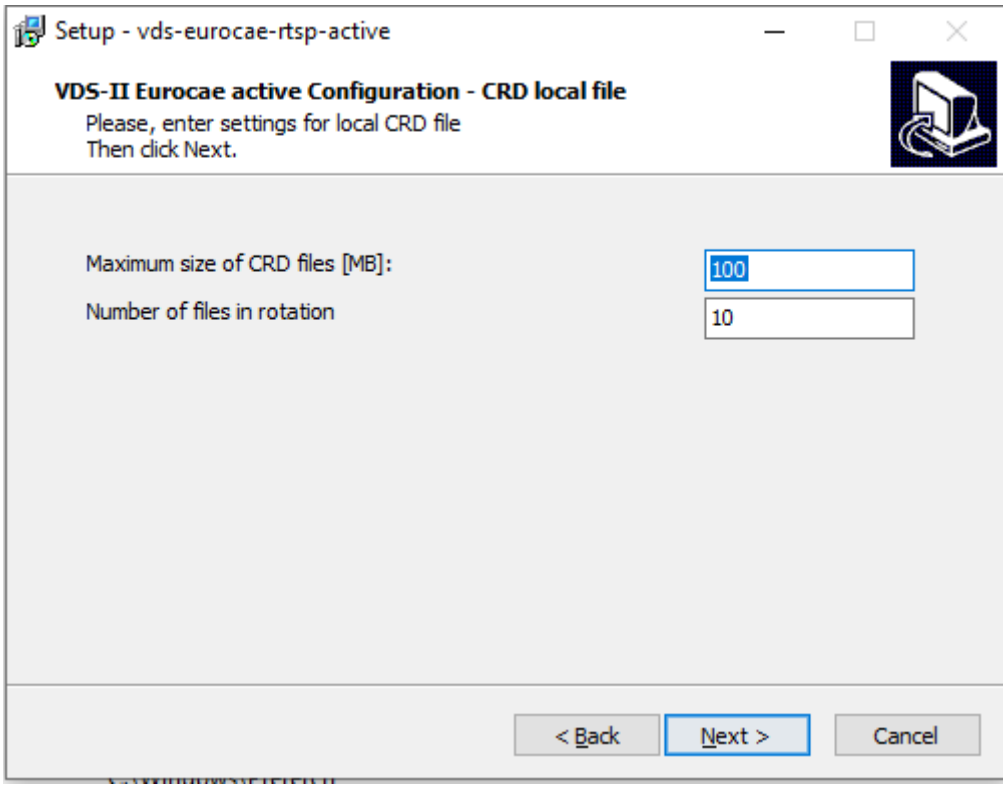
- wrapper.log VDS console output and java runtime console

12.9 crds.txt file

This will be generated only when the option **Store a CRD into local file** was chosen during the installation



The CRD Local file configurations should be done in this wizard



12.10 version.ini file

It contains the below contents

[Version]

Jar=5.0.6

Compiler=8

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

13 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.

13.1 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.IndexOutOfBoundException
- java.lang.ConcurrentModificationException

13.2 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 : XXXXXXX
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 *****