

# Installation & Configuration

**VDS-II RTP-Active**

## **Voice Recording System**

### **VC-MDX**

Version 1905

---

## Table of Contents

|          |                                     |           |
|----------|-------------------------------------|-----------|
| <b>1</b> | <b>Document History</b>             | <b>3</b>  |
| <b>2</b> | <b>General</b>                      | <b>4</b>  |
| 2.1      | Purpose                             | 4         |
| 2.2      | Acronyms and Abbreviations          | 4         |
| <b>3</b> | <b>Overview</b>                     | <b>5</b>  |
| <b>4</b> | <b>Installation</b>                 | <b>6</b>  |
| 4.1      | Prerequisites                       | 6         |
| 4.2      | Installation of VDS                 | 6         |
| 4.3      | Installed Files                     | 14        |
| 4.4      | Configuration of VC-MDx Recorder    | 16        |
| <b>5</b> | <b>VOX Detection</b>                | <b>20</b> |
| 5.1      | Operating principle                 | 20        |
| 5.2      | Configuration                       | 21        |
| 5.3      | Fields description                  | 21        |
| 5.4      | Input signal quality                | 22        |
| <b>6</b> | <b>Input Channels Configuration</b> | <b>23</b> |
| 6.1      | Structure of XML configuration file | 24        |
| 6.2      | Meaning of the attributes           | 24        |
| 6.3      | Mapping to Recorder fields          | 25        |
| 6.4      | Normative references                | 25        |
| 6.5      | XML Schema                          | 26        |
| <b>7</b> | <b>Static Routing</b>               | <b>28</b> |
| <b>8</b> | <b>Troubleshooting</b>              | <b>29</b> |
| 8.1      | Exceptions                          | 29        |
| 8.2      | ERROR log messages                  | 30        |
| 8.3      | Spring-config errors                | 33        |

## 1 Document History

| Version | Date       | Author | Changed         |
|---------|------------|--------|-----------------|
| 0.0     | 2016-12-09 | EV     | Initial Version |
| 1.0     | 2017-04-26 | EV     | Final release   |
| 1812    | 2018-12-20 | PE     | VC adapted      |
|         |            |        |                 |
|         |            |        |                 |

## 2 General

### 2.1 Purpose

This document describes VoiceCollect VDS-II RTP Active recording application. It covers the installation and the configuration of the VDS-II software. It does not cover the usage of the VC-MDx Recorder application nor of the operator's VC-MDx Workstation application, which have specific manuals.

### 2.2 Acronyms and Abbreviations

|      |  |
|------|--|
| AGC  | Automatic Gain Control                     |
| PCM  | Pulse Code Modulation                      |
| RTCP | Real Time Control Protocol (RFC3550)       |
| RTP  | Real-time Transport Protocol (RFC3550)     |
| UDP  | User Datagram Protocol (RFC768)            |
| VDS  | VoIP Decoding System (VoiceCollect)        |
| VEP  | VoIP Export Protocol (VoiceCollect)        |
| VOX  | Voice Operated eXchange (voice activation) |

### 3 Overview

VDS-II RTP Active software allows for the reception of VoIP audio which is sent as RTP streams by 3rd party systems such as radio gateways. The RTP streams are processed and then transferred in real time to VC-MDx Recorder for storage.

VDS provides a configurable number of Input Channels, each one corresponding to an independent RTP socket listening to a specific UDP port and dedicated to one audio source. The file that specifies the input channels can be modified on the fly. It can be located in VDS directory or at a remote place controlled by the 3<sup>rd</sup> party system.

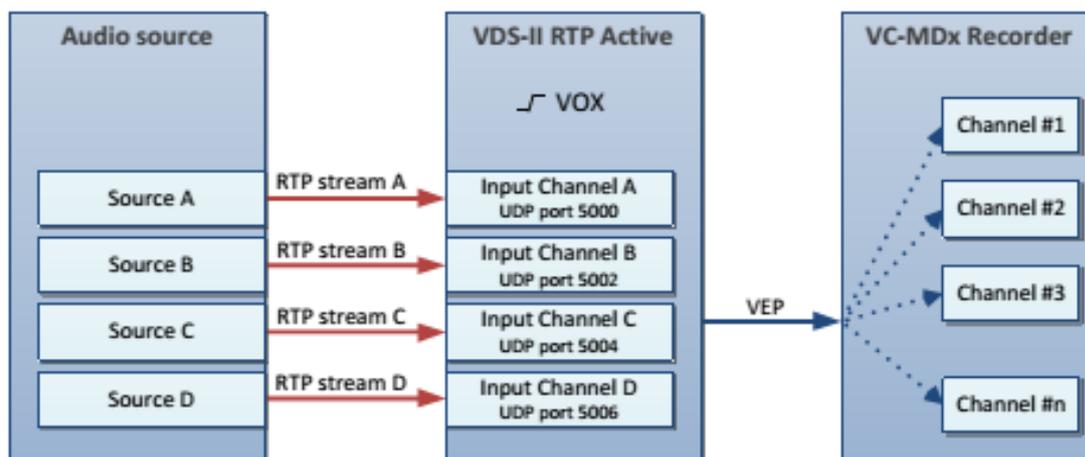
Supported audio codecs are G.711 uLaw/aLaw, G.723, G.726, G.728, G.729a, GSM FR, ILBC.

By configuration, when an incoming audio stream is detected on an input channel, it is routed either to the first idle recording channel of VC-MDx Recorder (Dynamic Routing) or to one dedicated channel of the Recorder (Static Routing).

VDS-II RTP Active can process unicast as well as multicast incoming streams.

By default, each incoming stream is considered a separate call. Recording begins when the stream begins and it ends after a short timeout when the stream ends.

An optional VOX feature allows for the detection of activity on incoming streams. For example if a sustained stream contains long periods of silence, recording can be triggered only when somebody is actually talking over the input channel.



## 4 Installation

### 4.1 Prerequisites

Before to proceed to VDS installation, it would be more convenient to define the number of input channels, the UDP ports to which audio streams will be sent to VDS and a "friendly" name for each input channel.

If the file for the input channel configuration is provided by a 3<sup>rd</sup> party application, the location of this file must be determined.

It is also useful to know if default dynamic routing will be used or if each input channel will be routed statically to a specific channel of the Recorder. If you plan to use Static Routing, then the Recorder must have at least as many channels as there are input channels.

**There is no limitation to the number of Input Channels, however the number of recorder channels is object of VoiceCollect licensing. Please, contact VoiceCollect Sales for details.**

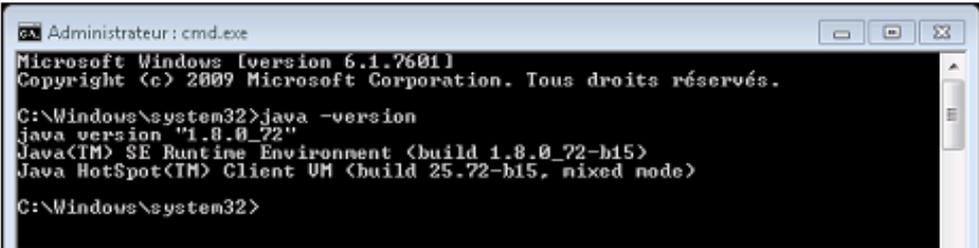
Ensure that communication between 3<sup>rd</sup> party systems and VoiceCollect applications will not be hindered by firewalls sitting in between. Incoming UDP packets must be allowed to the destination ports used by the input channels listeners of VDS. Similarly and in case VDS and the Recorder are on separate computers, incoming TCP ports 8510 and 8540 must be allowed for the connection from VC-MDx Recorder.

### 4.2 Installation of VDS

The VDS package is provided as a Windows installer named vds-rtp-active-4.1.3\_setup.exe (the version numbers can be different) to be installed on the computer. The screenshots below explain the most important steps.

**A 32-bit version of the Java Runtime Environment up to JRE 1.8 must already be installed.**

Use the following command to check if Java is already installed and which version it is



```
Administrateur: cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>java -version
java version "1.8.0_72"
Java(TM) SE Runtime Environment (build 1.8.0_72-b15)
Java HotSpot(TM) Client VM (build 25.72-b15, mixed mode)

C:\Windows\system32>
```

Locate the file vds-rtp-active-4.1.3\_setup.exe and double-click on the icon:

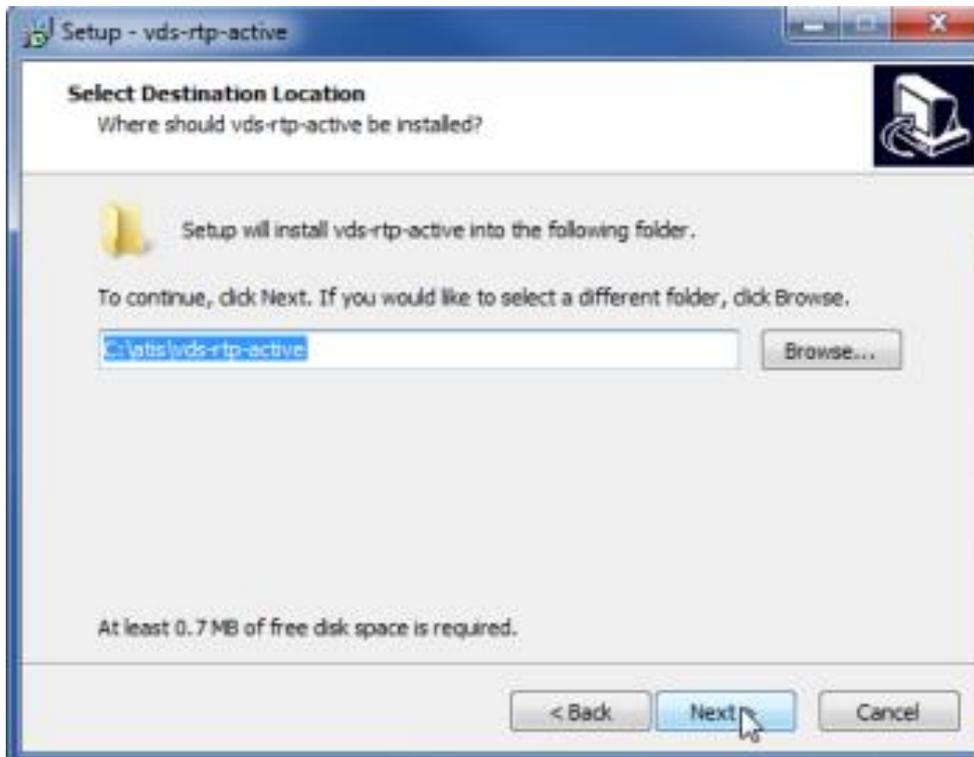


The version numbers may differ.

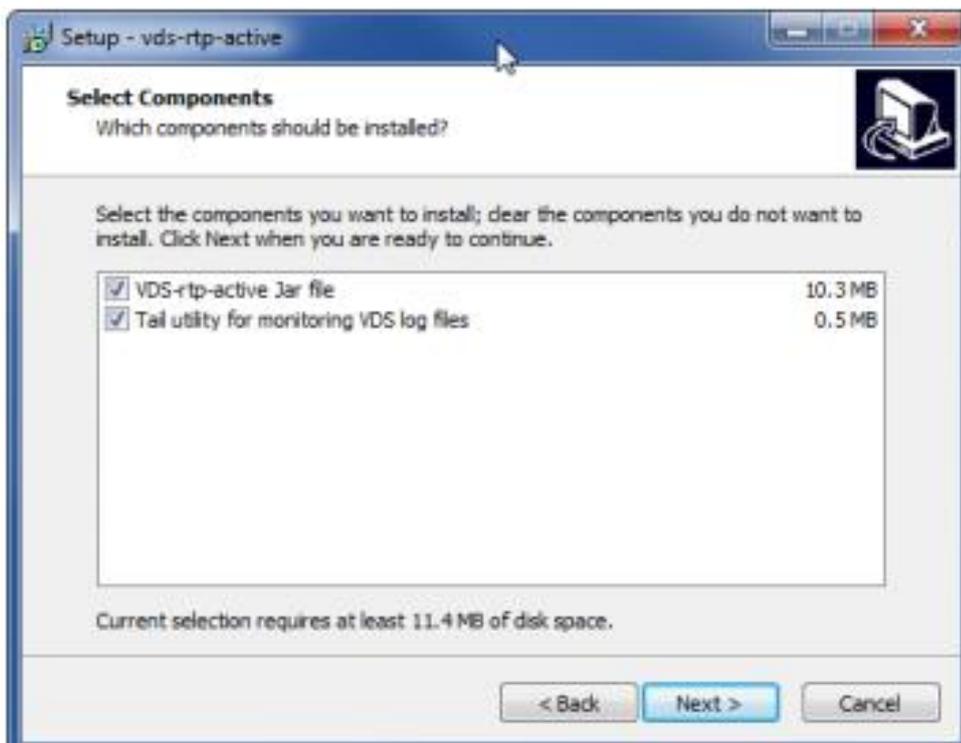
The following dialog box appears, press Next:



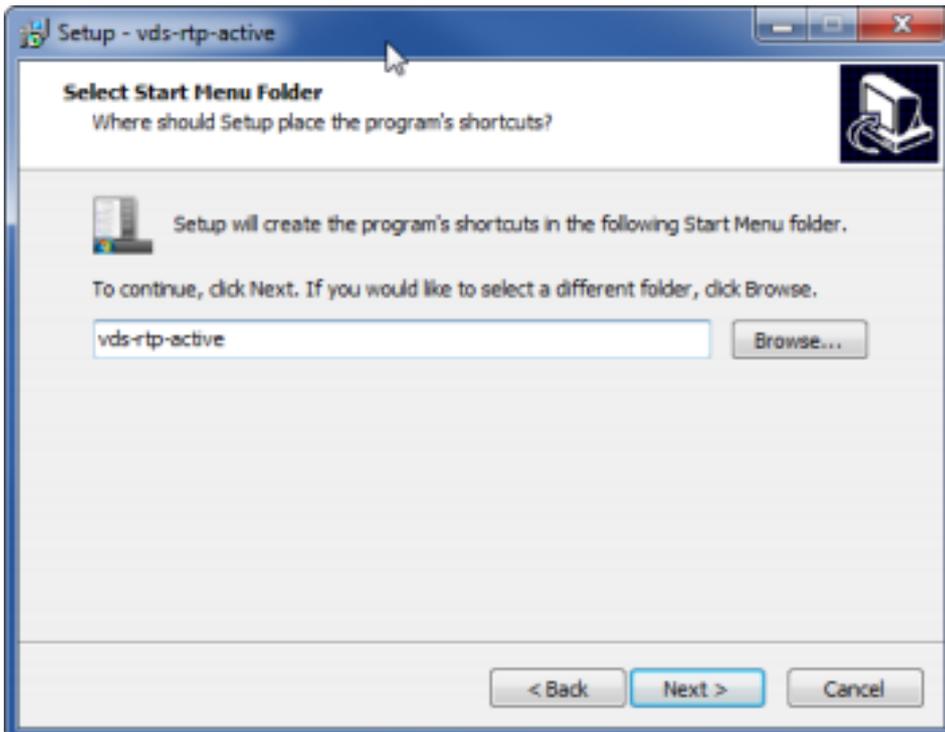
Select the directory where VDS-II will be installed in:



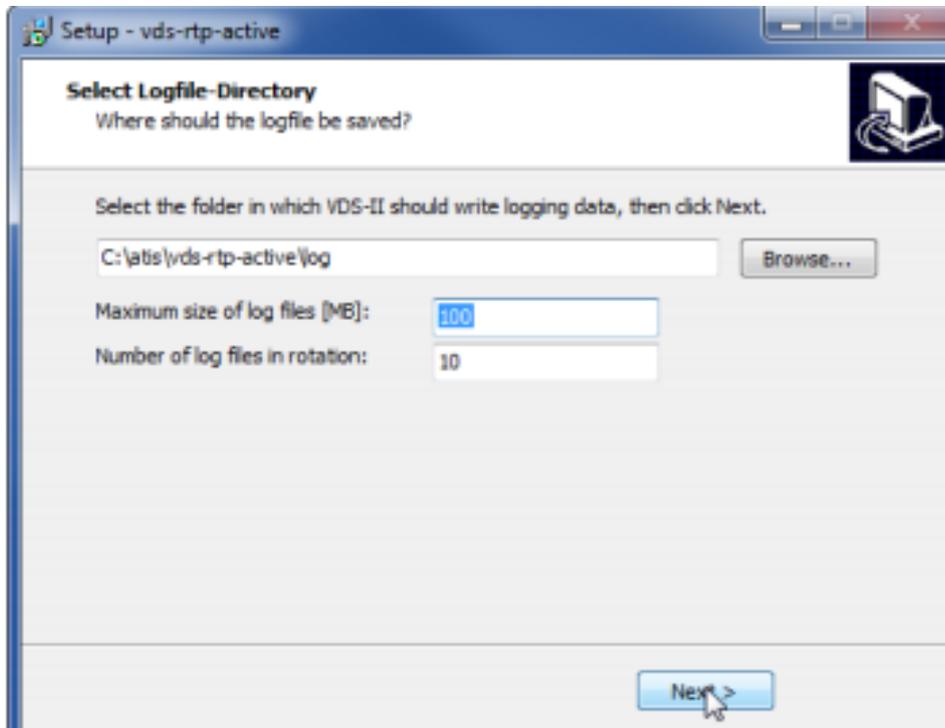
Select the components that will be installed:



Select the Start Menu folder:

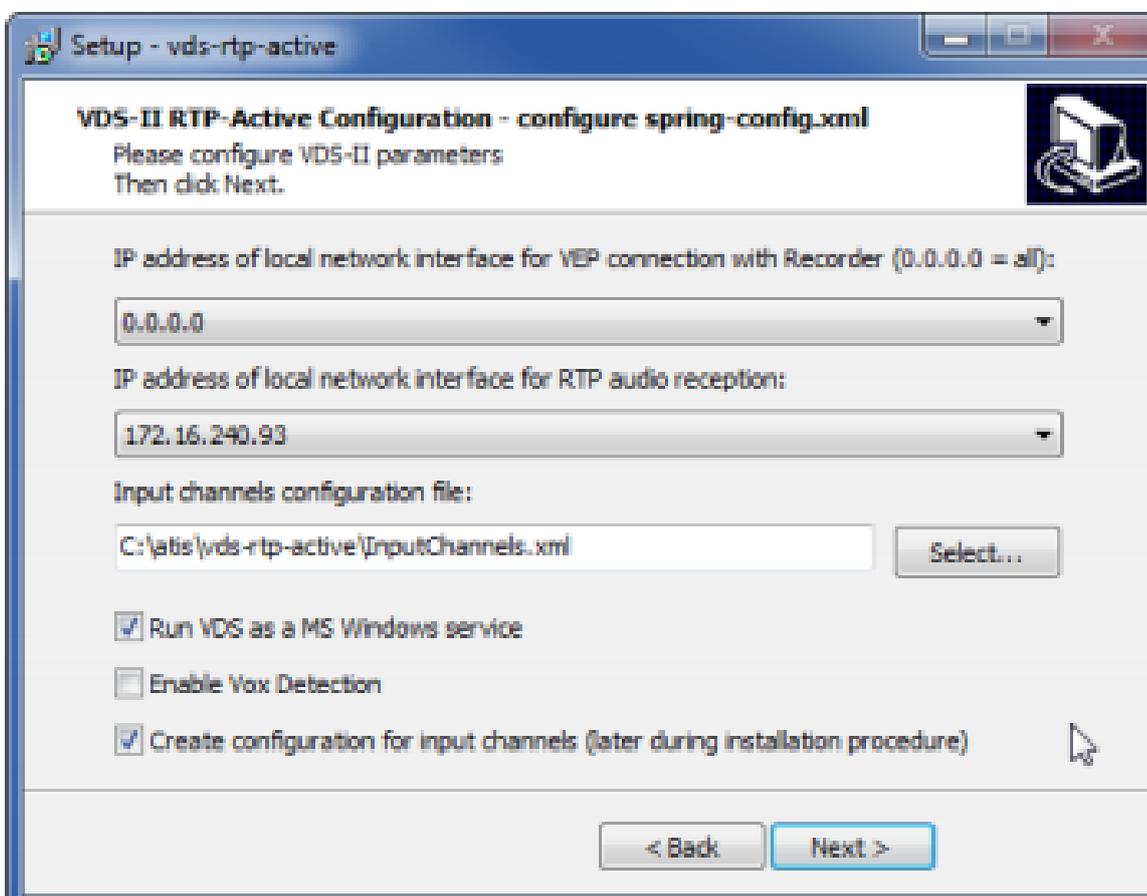


Select the location for the log files, as well as the size and the number of rotated log files:

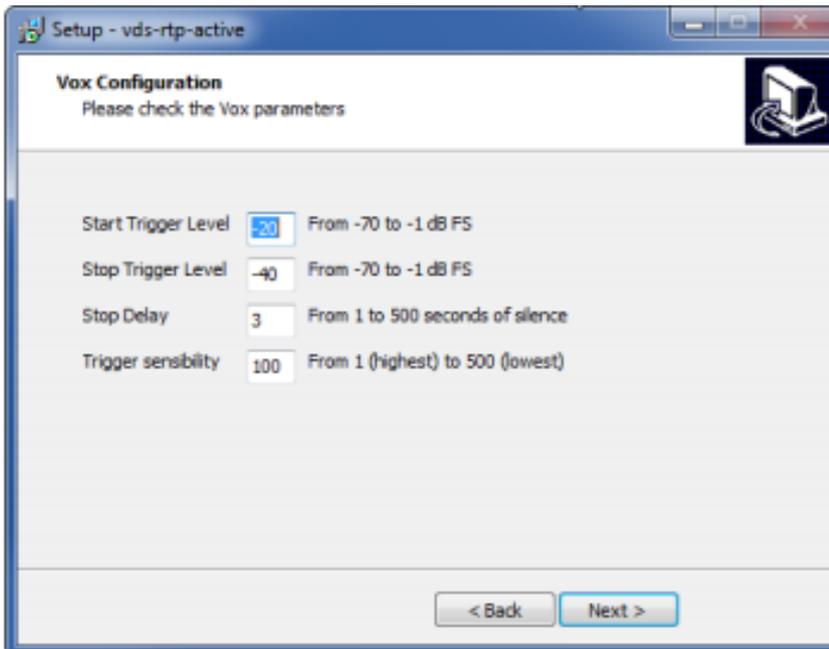


Next dialog allows specifying additional VDS parameters:

- Select the local network interface that will be used for the communication between VDS and VoiceCollect VC-MDx Recorder. Selecting 0.0.0.0 means that VDS will be listening to any local network interface. Please, ensure that there is no firewall blocking incoming connections from the Recorder.
- Select the local network interface that will be used for the reception of RTP audio streams from the external sources. Please ensure that there is no firewall blocking incoming packets from the external sources.
- If there is a pre-existing configuration file for the input channels, you can select it with the Select button or keep the default filename. This file will be overwritten if you then choose to create a new file with the option below in this dialog box.
- Check the option to run VDS in the background as a MS Windows Service (default).
- Check VOX detection on incoming streams if desired (See section VOX Detection for details)
- If desired, check the option to create a configuration file for the input channels. If the file already exists, then it will be overwritten.

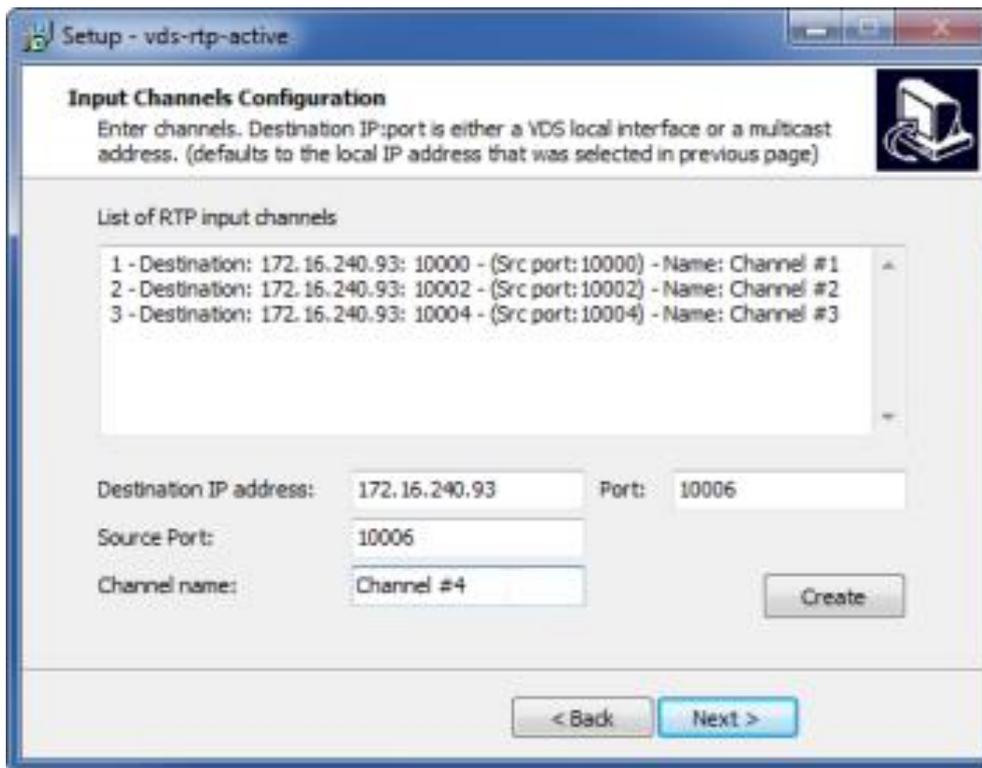


Next dialog appears if the VOX Detection option was selected. It allows for the fine adjustment of VOX settings such as audio level of the start and stop triggers for recording. Default values fit well for most input signals (See section VOX Detection for details).



Next dialog is displayed if the option for the creation of the Input Channels configuration file was selected. It offers the possibility to add RTP channels for the external audio sources and then save the list in the input channel configuration file.

- Each input channel entry consists of a destination IP address and destination UDP port on the VDS computer. By default the address is the one that was entered in previous dialog box. In case of multicast audio streams, then the multicast IP address must be entered. Note that port attribution follow standard rule which is RTP ports use even numbers (see section Input Channel Configuration for details).
- The source port is the one which is used on the remote source for a given source. Note that this value is currently not used by VDS. If it is undefined, you can set the same value as for the destination port.
- In the channel name field, you can enter a friendly name that will be used to identify the audio source. This field will appear with the call related data of each recording



Note that the channel configuration file is checked regularly (each 30 seconds) by VDS during the run time. It can be modified at any time and VDS will reload the new configuration.

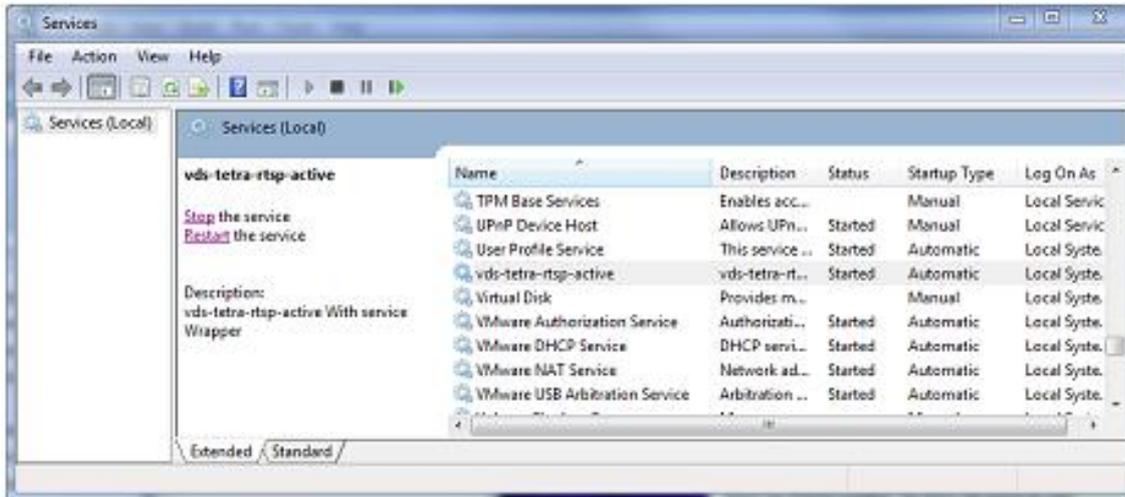
Last step indicates the completion of the installation procedure. The configuration file `springconfig.xml` in VDS' directory is adjusted to current installation and the program will start automatically.



If something later needs to be changed in VDS configuration, for example the filename and location of the input channel configuration or VOX threshold values, you can edit `spring-config.xml`

**Please, be prudent if you intend to modify `spring-config.xml` which is a sensitive file!**

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



In case of problems, you can also check the log files in the log directory of the program. See section [Troubleshooting](#) below.

## 4.3 Installed Files

### Spring-config.xml

This file describes the VDS software, some parameters can be modified manually but we recommend extreme care when editing that file.

### 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).

### Runtime.xml

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

### Batch files

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

- install-vds-rtp-active-service.bat      Install the service
- remove-vds-rtp-active-service.bat      Remove the service
- start-vds-rtp-active-service.bat      Start the service
- stop- vds-rtp-active-service.bat      Stop the service

These files are just there for convenience if you want to start/stop VDS without the need to go through the MS Service Manager.

**Dll Files**

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

**Wrapper files**

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

**Jar file**

This is the main VDS software

**Log files**

Two log files are created into the directory chosen during installation

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

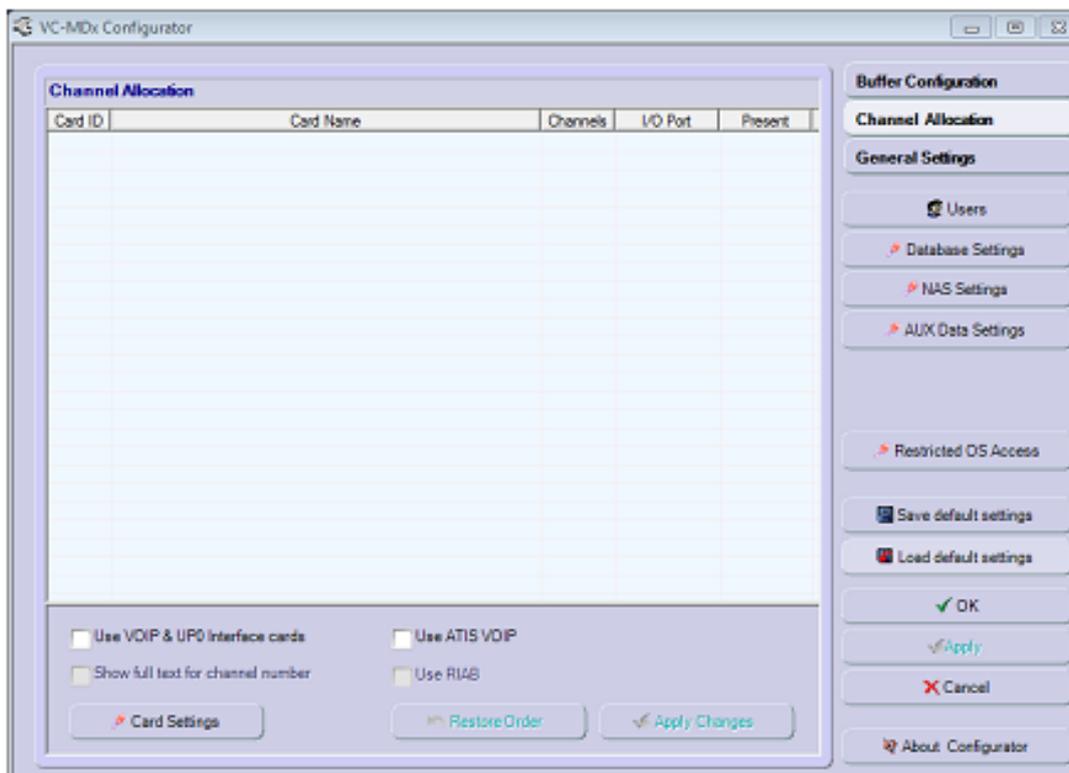
## 4.4 Configuration of VC-MDx Recorder

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.

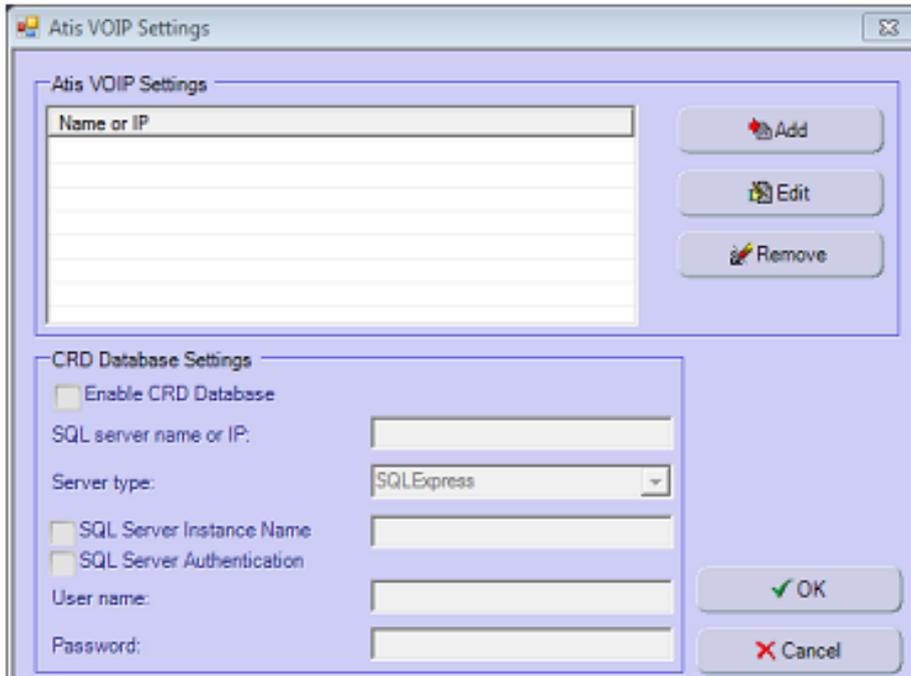
For configuring the Recorder, the VDS application must be up and running.



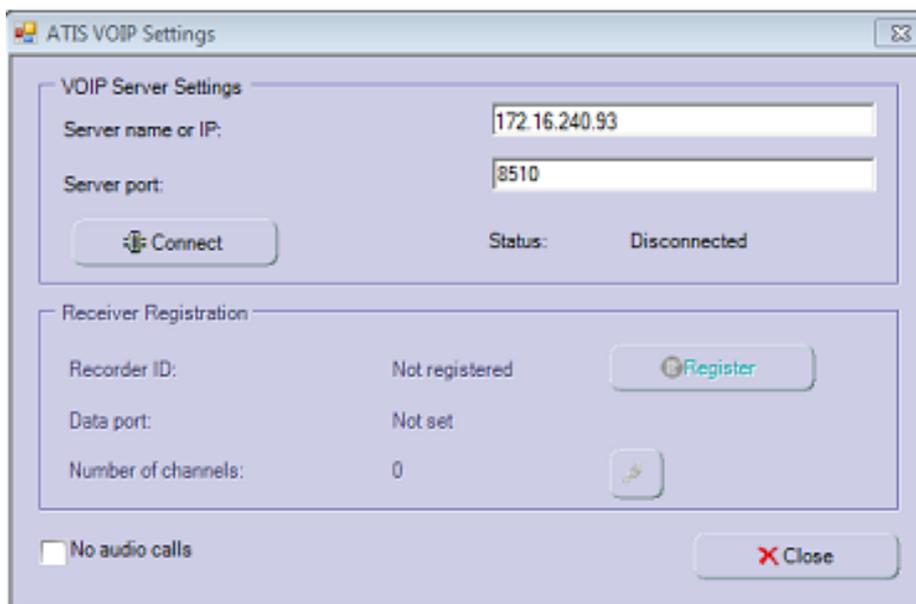
The following dialog box appears. Check the Use ATIS VOIP, a new dialog will pop up:



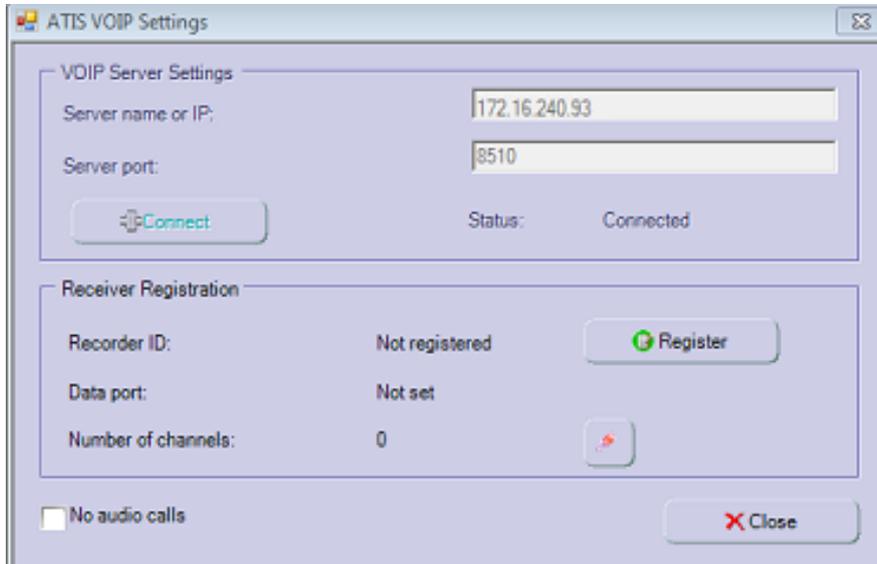
The new dialog displays a list of connections to VDS applications (currently empty). Press the Add button:



Next dialog displays parameters for the VEP connection to VDS. In the Server name or IP field, enter the IP address of VDS interface (See Installation of VDS). Then press the Connect button:



Once the connection is established, press the Register button:

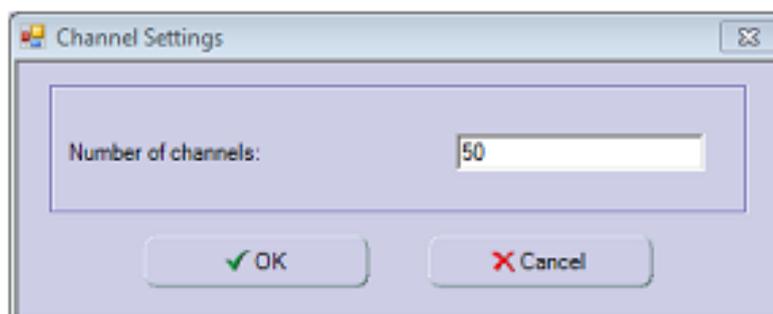


Note: The No audio calls checkbox allows for keeping records of calls without audio, for example rejected or discarded calls. As this cannot happen with VDS-II RTP Active, you can leave it unchecked.

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

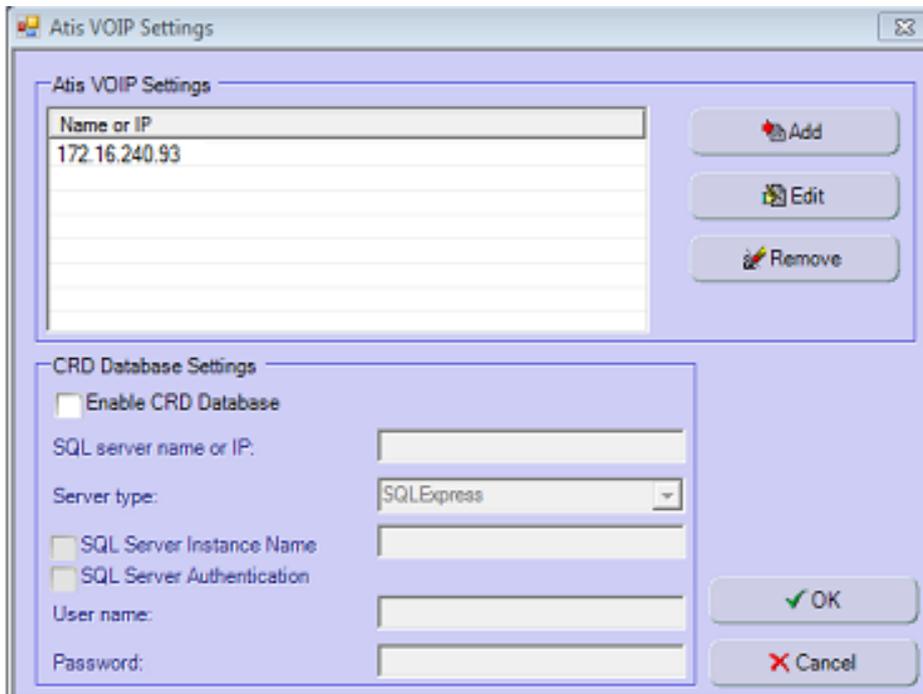


Following dialog appears where you can enter the number of channels:



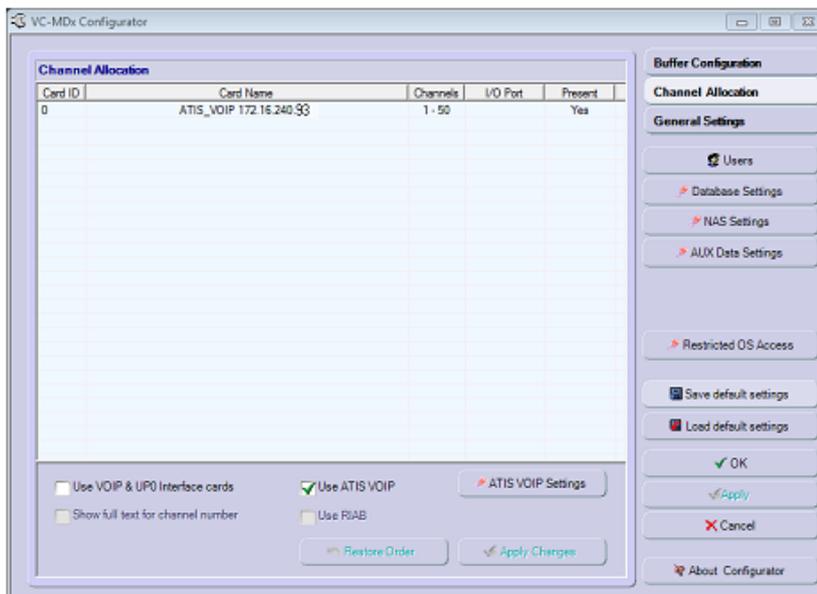
The number of channels is object of VoiceCollect licensing. Please, contact VoiceCollect Sales for details.

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



The CRD Database is not used in current context. Leave the checkbox unchecked. Press the OK button to finish VDS registration.

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



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

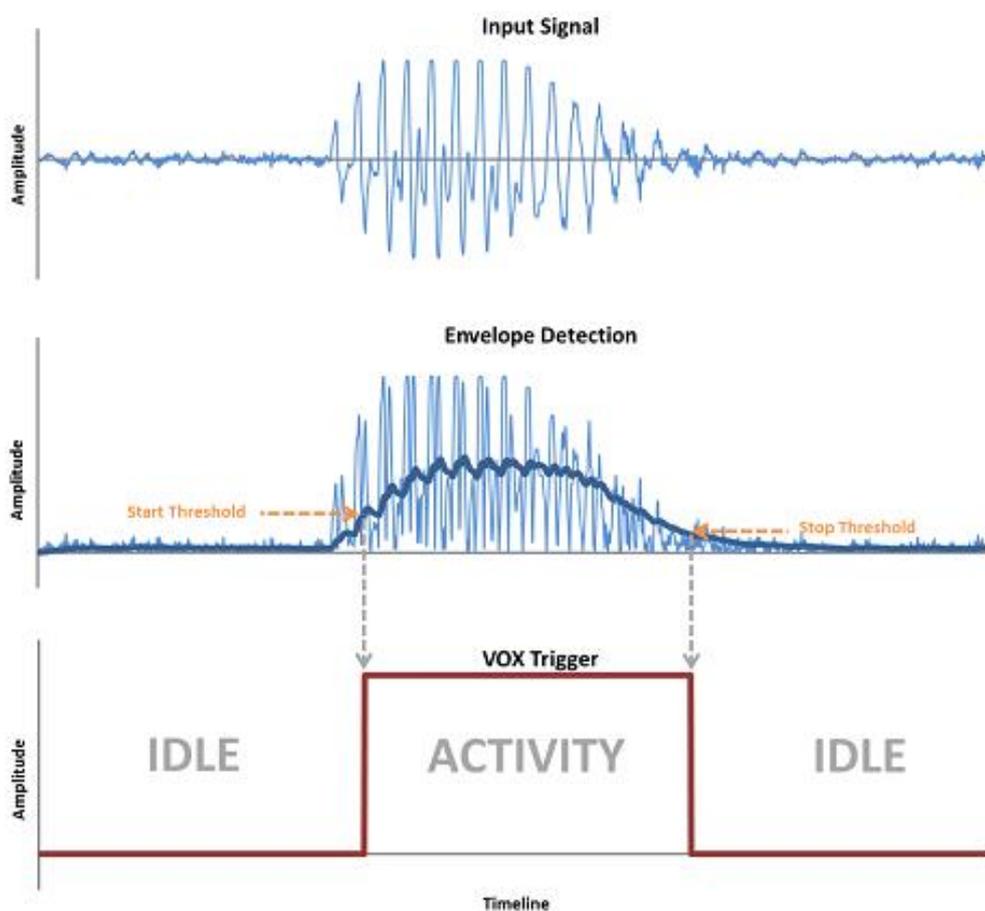
**VDS and the Recorder are now ready to use.**

## 5 VOX Detection

VDS-II RTP Active 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.

### 5.1 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.

## 5.2 Configuration

To activate the VOX function, open the file `spring-config.xml` in VDS directory. In the list of properties under "ioModules", change the value of the "vox" property to "true". This will activate VOX with default values which work for most cases.

If you need to adjust some of the VOX parameters, insert or modify following properties below the "vox" property:

```
<property name="vox" value="true" />
<property name="startThresholdInDb" value="-20" />
<property name="stopThresholdInDb" value="-40" />
<property name="stopDetectionInSec" value="3" />
<property name="smoothValue" value="100" />
```

## 5.3 Fields description

| VOX property              | Description                                    | Values                                    | Remarks   |
|---------------------------|--|---|---|
| <b>vox</b>                | Enable/disable vox function                    | true/false<br><i>Default: false</i>       | VOX is deactivated by default   |
| <b>startThresholdInDb</b> | Trigger level for activity detection           | -70 to -1 dB FS<br><i>Default: -20 dB</i> | If not present or set to 0, default will be used. Must be set higher than stopThresholdInDb   |
| <b>stopThresholdInDb</b>  | Trigger level for silence detection            | -70 to -1 dB FS<br><i>Default: -40 dB</i> | If not present or set to 0, default will be used. Must be set lower than startThresholdInDb   |
| <b>stopDetectionInSec</b> | Hang time for stop detection                   | 1 to 20 s<br><i>Default: 3 s</i>          | Recording will be stopped only if signal stays under stop threshold for that time. Allows for short speech pauses during recording.                                   |
| <b>smoothValue</b>        | Degree of smoothing in signal envelop detector | 1-500<br><i>Default: 100</i>              | The higher, the less sensible to changes (e.g. if too many starts get triggered by short audio cracks, increase that value). Values too high decrease responsiveness. |

## 5.4 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 Input Channels Configuration

The list of RTP listeners (input channels) is defined in an XML file that can be modified on-the-fly, for example by a 3<sup>rd</sup> party application. VDS checks this file for changes every 30 seconds.

For each Input Channel the configuration file will contain the destination IP address and UDP port as well as a human-friendly name that will be displayed on the Recorder in the "User ID" field.

The destination IP can be either an IP address of the VDS computer or a multicast address, in which case VDS will initiate an IGMP "join group" message when opening the multicast socket.

The path and filename of the Input Channels configuration file is set in VDS configuration filespring-config.xml. By default the installer proposes to create the file in VDS directory but the file can be located anywhere, for example on a remote computer (SMB share) where both VDS and the customer can access it. The file can also be accessed at remote locations using HTTP and FTP protocols.

Example of paths as they can be set in spring-config.xml:

- C:\atis\vds-rtp-active\InputChannels.xml
- \\SERVER\share\config\VDS-II\VDS-II\InputChannels.xml
- http://www.domain.local/config/VDS-II/InputChannels.xml
- ftp://user:password@www.domain.local/config/VDS-II/InputChannels.xml

Abstract of spring-config.xml:

```
<bean name="rtpActiveModule"
class="com.atissystems.recorder.vds.protocol.rtpactive.IoModuleRtpActive"
init-method="init">
  <property name="localInterfaceIpAddr" value="172.16.240.93" />
  <property name="channelConfigFilename" value="d:\InputChannels.xml" />
  <property name="moduleId" value="12234" />
  <property name="recordingDataProcessor" ref="recordingDataProcessor" />
</bean>
```

The file name and location are set in bean 'rtpActiveModule', property 'channelConfigFilename'.

The property 'localInterfaceIpAddr' in same bean specifies the IP address of VDS that will be used by default in listening sockets for the RTP channels.

## 6.1 Structure of XML configuration file

The structure is defined by the XML schema which is provided. Resulting Input Channels configuration file will look as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rtpChannelList>
  <rtpChannel destPort="10000" destIP="172.16.240.93" transport="udp"
    srcPort="5000" name="Channel #1"/>
  <rtpChannel destPort="10002" destIP="172.16.240.93" transport="udp"
    srcPort="5002" name="Channel #2"/>
  <rtpChannel destPort="10004" destIP="172.16.240.93" transport="udp"
    srcPort="5004" name="Channel #3"/>
  <rtpChannel destPort="10006" destIP="172.16.240.93" transport="udp"
    srcPort="5006" name="Channel #4"/>
  <rtpChannel destPort="10008" destIP="172.16.240.93" transport="udp"
    srcPort="5008" name="Channel #5"/>
  <rtpChannel destPort="10010" destIP="172.16.240.93" transport="udp"
    srcPort="5010" name="Channel #6"/>
  <rtpChannel destPort="10012" destIP="172.16.240.93" transport="udp"
    srcPort="5012" name="Channel #7"/>
  <rtpChannel destPort="10014" destIP="172.16.240.93" transport="udp"
    srcPort="5014" name="Channel #8"/>
  <rtpChannel destPort="20000" destIP="224.224.224.6" transport="udp"
    srcPort="20000" name="Mcast Channel #9"/>
  <rtpChannel destPort="80010" destIP="228.50.60.7" transport="udp"
    srcPort="80010" name="Mcast Channel #10"/>
</rtpChannelList>
```

The order of the attributes is not important, but all attributes must be present

## 6.2 Meaning of the attributes

- The 'destPort' attribute specifies which port will be used in VDS listening socket for this channel. The rule in RFC3550 that specifies that RTP ports are even numbers and their associated RTCP ports must be next odd numbered port applies: only even port values are accepted. Minimum allowed port number is 1024 (IANA local "System Ports" are protected).
- The 'destIP' attribute usually contains the IPv4 address of VDS for the listening sockets, as defined in the file 'spring-config.xml', or it can contain "0.0.0.0" (any). If left empty, default address will be used. The IPv4 dotted notation formatting must follow the rules from the IETF (see also RFC3986).
- If the RTP stream is multicast, then the multicast address must be provided in this field. In this case, the address in spring-config.xml will be used for sending the IGMP "join group" packet through the correct physical interface.
- The 'transport' attribute specifies if the stream will be transported with TCP or with UDP (In current versions, only UDP transport is provided).

- The 'srcPort' attribute indicates the port used by the source of the RTP stream. It is currently not used in VDS and any even port is allowed.
- The 'name' attribute represents a friendly name for the channel and it will be displayed in the VC-MDx Recorder for each recorded call. It can be up to 30 characters long.

## 6.3 Mapping to Recorder fields

For each record, attributes described above and actual source IP address and port will be mapped to VC-MDx Recorder fields as follows:

| VDS call data   | VC-MDx Recorder |
|-----------------|-----------------|
| name            | User ID         |
| srcIP:srcPort   | Calling Number  |
| destIP:destPort | Called Number   |

## 6.4 Normative references

IETF: RFC3550 "RTP: A Transport Protocol for Real-Time Applications"  
(<http://tools.ietf.org/rfc/rfc3550.txt>)

IETF: "Textual Representation of IPv4 and IPv6 Addresses"  
(<https://tools.ietf.org/id/draft-main-ipaddr-text-rep-02.txt>)

IETF: RFC3986 "Uniform Resource Identifier (URI): Generic Syntax"  
(<http://tools.ietf.org/rfc/rfc3986.txt>)

IANA: "Service Name and Transport Protocol Port Number Registry"  
(<http://www.iana.org/assignments/service-names-port-numbers/service-names-portnumbers.xhtml>)

## 6.5 XML Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="channelNameType">
    <xs:annotation>
      <xs:documentation>Name of the channel</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="30"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="transportType">
    <xs:annotation>
      <xs:documentation>Packet transport (TCP/UDP)</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="udp"/>
      <xs:enumeration value="tcp"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ipAddrType">
    <xs:annotation>
      <xs:documentation>IPv4 address (formatting rules from RFC3986)</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:pattern value="((([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])\.)\.)\{3}([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="srcPortType">
    <xs:annotation>
      <xs:documentation>Source port number (0-65535).</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:integer">
      <xs:minExclusive value="0"/>
      <xs:maxInclusive value="65535"/>
      <xs:pattern value=".*[02468]"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="destPortType">
    <xs:annotation>
      <xs:documentation>Destination port number (1024-65535). IANA System Ports (0-1023) are forbidden.</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:integer">
      <xs:minExclusive value="1023"/>
      <xs:maxInclusive value="65535"/>
      <xs:pattern value=".*[02468]"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="rtpChannelList">
    <xs:complexType>
      <xs:sequence>
```

```
<xs:element name="rtpChannel" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>Definition for an RTP source channel</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="name" type="channelNameType" use="required"/>
    <xs:attribute name="srcPort" type="srcPortType" use="required"/>
    <xs:attribute name="transport" type="transportType" use="required"/>
    <xs:attribute name="destIP" type="ipAddrType" use="required"/>
    <xs:attribute name="destPort" type="destPortType" use="required"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>
```

## 7 Static Routing

Normally, VDS is set to run with dynamic routing, meaning that when there is activity on an Input Channel, it will take the first free channel of the Recorder to record the call.

However, it is possible to set VDS to statically assign specific Input Channels to specific recording channels.

To use Static Routing, you must create or edit the XML file named `routing-configuration.xml` in VDS directory. This file must have the following structure:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<routingTable>
  <location>Any</location>
  <recorderList>
    <recorder name="R0001">
      <channelList>
        <channel id="1" filter="Channel #1"/>
        <channel id="2" filter="Channel #2"/>
        <channel id="3" filter="Channel #3"/>
        <channel id="4" filter="Channel #4"/>
        <channel id="5" filter="Channel #5"/>
        <channel id="6" filter="Channel #6"/>
        <channel id="7" filter="Channel #7"/>
        <channel id="8" filter="Channel #8"/>
        <channel id="9" filter="Mcast Channel #9"/>
        <channel id="10" filter="Mcast Channel #10"/>
        <channel id="11" filter="**"/>
        <channel id="12" filter="**"/>
        <channel id="13" filter="/"/>
        <channel id="14" filter="/"/>
      </channelList>
    </recorder>
  </recorderList>
</routingTable>
```

Each recording channel entry is assigned a filter which compares to the friendly name of the Input Channels. On new call, if there is a match in the list, the recording will be assigned to the channel with corresponding ID.

"\*" 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, the new routing will be taken into account on next successful call.

## 8 Troubleshooting

As the VDS 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 files for specific errors.

### 8.1 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 as soon as possible:

**java.lang.NullPointerException**

**java.lang.OutOfMemoryError**

**java.lang.IndexOutOfBoundsException**

**java.lang.ConcurrentModificationException**

## 8.2 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 messages are logged with the format "Exception + message". Most messages should be self-explanatory.

Here are some examples of some of these error messages.

When the errors are coming from the 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 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.

**Note: 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, channel number = possible simultaneous recording)

When the VDS need a channel it tries to acquire 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 the file *routingconfiguration.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.

## 8.3 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.

**Caused by: java.lang.UnsatisfiedLinkError: C:\Atis\Test\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 on a 64 bits java runtime environment. De-install the java runtime 64bits and install a 32 bits version of the java runtime (JRE)

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