

VOICEMEETER

Real Time Virtual Audio Mixer for Windows



USER MANUAL

OFFICIAL WEBSITE
www.voicemeeter.com

This Document is a Step By Step Manual.
Thanks to follow every step until getting the wanted configuration.

USE CASES:

- How to talk and send music in the same time on Skype or Google Voice?
- How to manage 2 headsets on Skype or Google voice?
- How to record Conference-Call in 8 tracks for post production?

INSTALLATION:

Run setup program and follow instructions.

Reboot after installation or de-installation

IMPORTANT NOTE: If you do not reboot after de-installation, virtual audio drivers might not be completely de-installed and prevent you to install new version correctly. If it happens, you will be obliged to de-install Voicemeeter's Virtual audio device's) manually.
(by the system / device manager windows system procedure).

ONLINE SUPPORT

<https://forum.vb-audio.com/viewforum.php?f=6>

Fair Trade, Affordable For Everyone

Voicemeeter is a donationware, free to download and free to use!
It allows you to adjust the license price according to your means or usage.

Thanks for your participation and support!

<https://shop.vb-audio.com>

Windows XP, VISTA, WIN7, WIN8, WIN8.1, WIN10 32/64 bits (MME, DX, WDM/WASAPI, KS, ASIO).
www.voicemeeter.com / www.vb-cable.com

Table of Content

INTRODUCTION:	6
Voicemeeter as Universal Virtual Audio Device:.....	7
Voicemeeter as Audio HUB and Next Gen Audio Engine:.....	8
Audio Mixing Console Basis:.....	9
STEP 0: Quick Startup	10
Remarks in case of problem:	10
STEP 1: Select main audio Output Device	11
Windows Control Panel / Sound Properties to configure Device Sound Quality:.....	11
STEP 2: Select an Input Device.	12
Monitor Input in Real Time	13
WINDOWS AUDIO INPUT CONFIGURATION WARNING	13
STEP 3: Use Virtual Input as default playback device.....	14
Virtual Input is Multi-Channel !	15
Understand Voicemeeter Virtual I/O in one picture.	16
STEP 4: Send the Mix Output to Skype Input.	17
STEP 5: Enjoy new Audio Controls.....	18
INTELLIPAN 3D PANEL: The Binaural effect.....	18
Audibility control & equalizer.....	19
WINDOWS CONFIGURATION WARNING WITH VOIP	19
STEP 6: Connecting ASIO Application to Voicemeeter.	20
Special Routing Options on Output BUS.	21
M.C. Means “Mute Center”	21
ASIO Routing Capabilities.	22
Menu for Extra functions	23
CASE STUDY #1	25
CASE STUDY #1 – STEP 1.....	26
CASE STUDY #1 – STEP 2.....	27
CASE STUDY #1 – STEP 3.....	28
CASE STUDY #2	30
CASE STUDY #2 – STEP 1.....	31
CASE STUDY #2 – STEP 2.....	31
CASE STUDY #3	33
VBAN: VB-Audio Network	35
Configure VBAN Audio Stream.....	36
Send audio stream to other computer:.....	36

Receive audio stream from other computer:.....	36
Additional VBAN-MIDI or Command stream.....	37
VBAN2MIDI application.....	37
MACRO Buttons.....	39
Configuring MACRO Buttons	40
Button Color	41
Voicemeeter Remote Requests.....	41
Input Strip Parameters:	42
BUS Parameters:.....	42
Special functions to make timed fade in/out:.....	43
System Settings Option:	43
Special Commands:	44
Button command for button interactions.....	44
Wait command to create sequence of requests.....	44
Load Button map.....	45
VBAN Options:.....	46
AUTO Ducking (Trigger):.....	47
System Functions (to send command to Windows):.....	48
System Execute.....	48
System KeyDown / KeyUp / KeyPress	49
List of Key Name:.....	49
Send M.I.D.I. Message	50
Send VBAN-MIDI or VBAN-TEXT (Voicemeeter script).....	52
Send MIDI command through VBAN Stream	53
Send Voicemeeter TEXT Request through VBAN Stream.....	53
Manage your lights Network in DMX 512	54
Voicemeeter Remote API (for developer only)	55
System Settings / Options	57
ASIO Driver support:.....	58
Getting Optimal Latency:	59
LATENCY WARNING:.....	59
M.I.D.I. Mapping:.....	60
Specifications:.....	61
Voicemeeter I/O Diagram:	62
KNOWN ISSUES / RECOMMENDATIONS	63

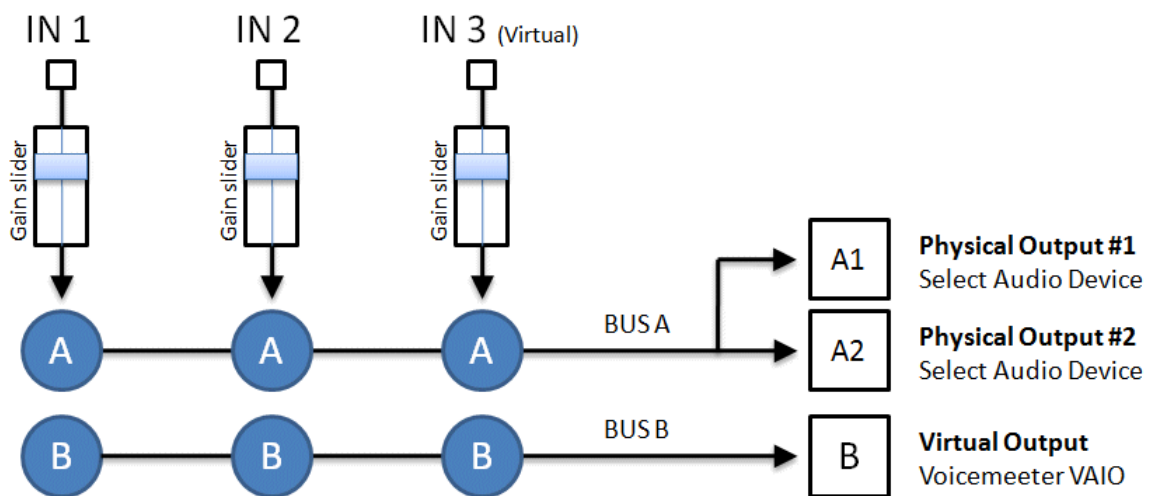
No sound anymore on my computer!	63
Voicemeeter does nothing if you don't use it	63
Audio Stream becomes bad (sound is choppy, discontinued):	63
Output A1 and A2 are not exactly synchronized	63
Voicemeeter Audio Stream can be stopped (no more sound, no meter display).	63
Audio Stream is stopped or corrupted after working well several hours.	63
Command Line Installation:.....	64
Voicemeeter Command Line:.....	64
Registry Parameters:	65

INTRODUCTION:

Voicemeeter is a virtual mixing console able to manage 3 audio inputs (2 physicals and 1 virtual) and 3 audio outputs (2 physicals and 1 virtual) through 2 busses A & B.



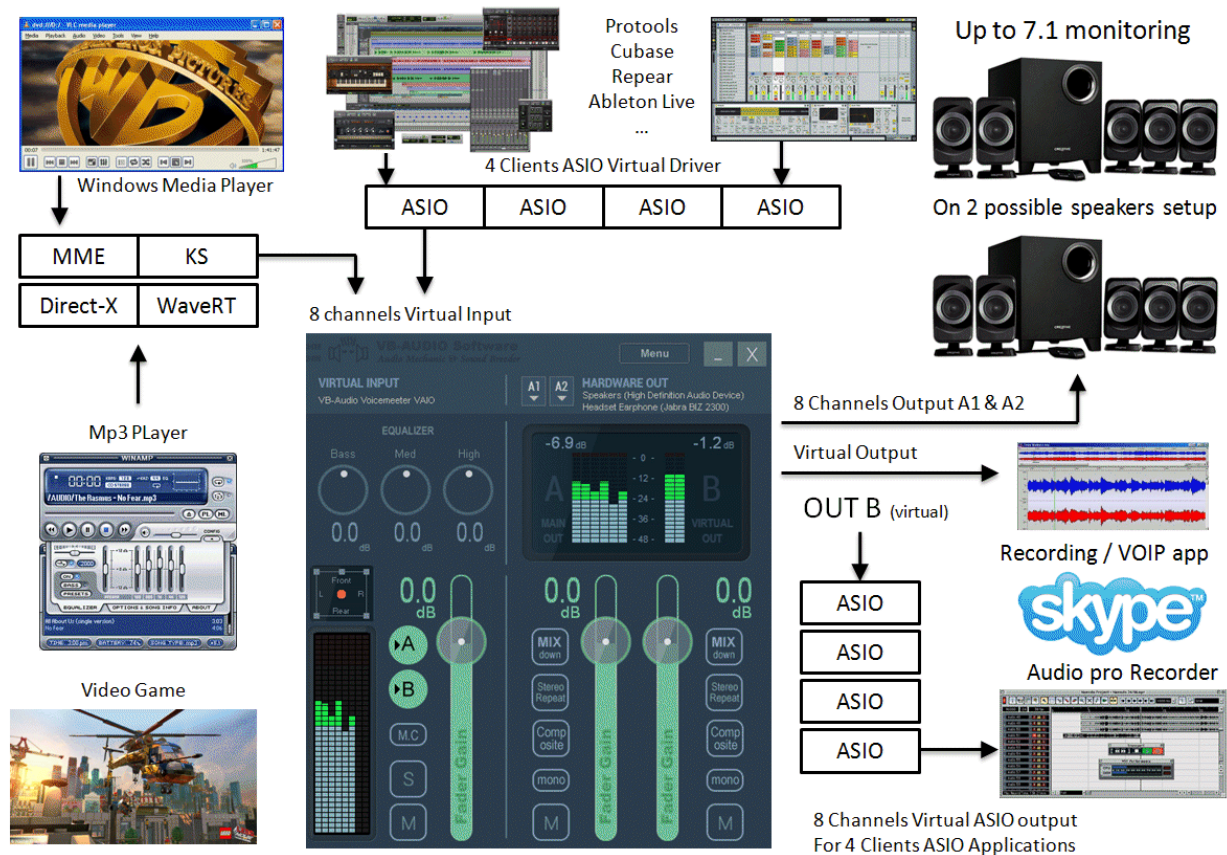
As Mixing Console, Voicemeeter offers large amount of use case combinations: To mix in real time your voice with your music, your movie, your video game or web radio and share it on internet through VOIP applications. Voicemeeter is also able to let you manage 2 Headsets and create new VOIP experiences ...



Voicemeeter General Diagram (3 inputs / 2 Buses Mixing Console)

Voicemeeter as Universal Virtual Audio Device:

Voicemeeter offers Virtual Audio Point on Input #3 and on BUS B (Output B). This Virtual Audio I/O (VAIO) supports all possible Audio Interfaces and allows connecting any audio applications, including audio pro DAW or musical instrument working with ASIO devices.

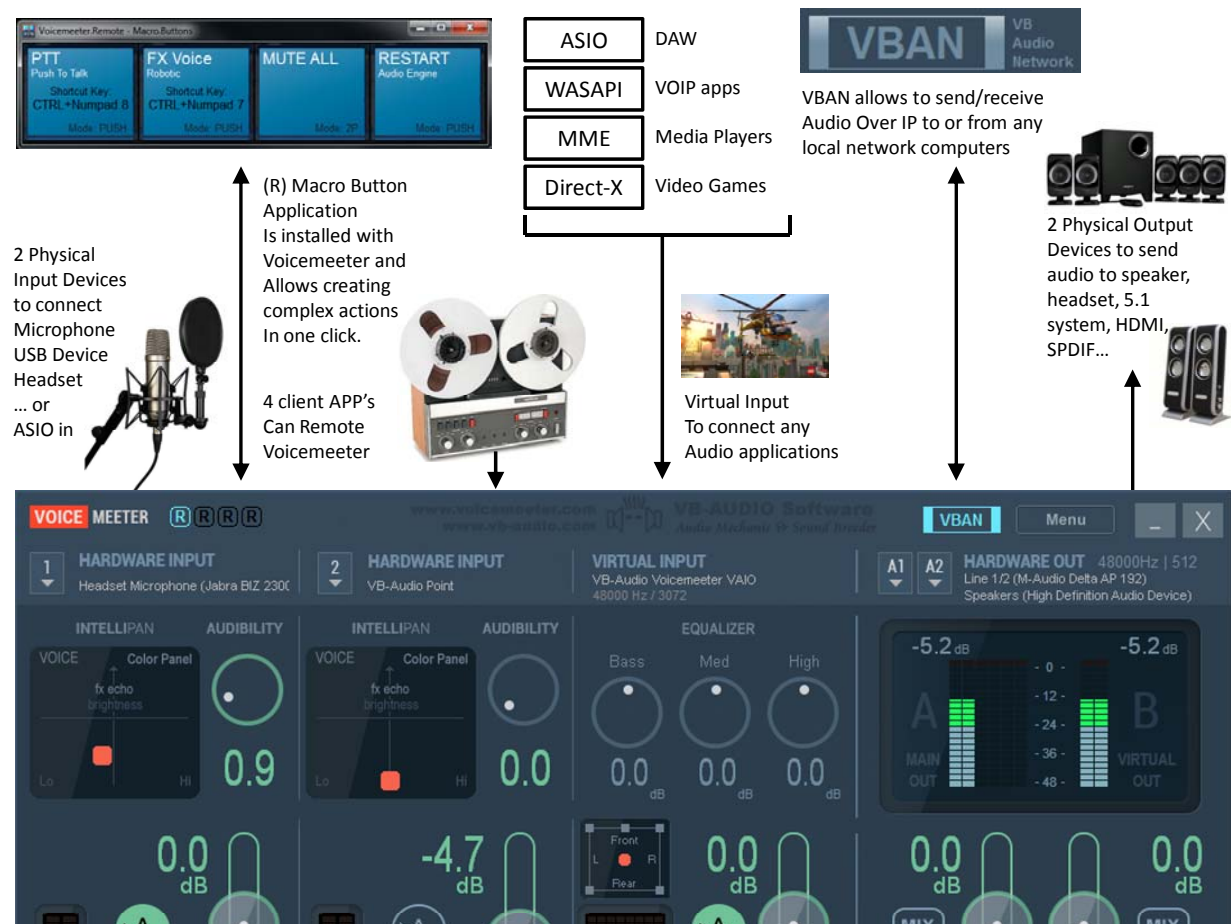


Audio interfaces type supported by Voicemeeter Virtual audio I/O (Voicemeeter VAIO).

Interface Type	Description
MME	The MME API or the Windows Multimedia API (also known as WinMM) was the first universal and standardized Windows audio API. This audio interface type is expected to work with any audio device but with latency time possibly around 100ms.
WDM	That we call WDM (Windows Driver Model) audio interface in Voicemeeter is handled by WASAPI : the Latest Microsoft Audio Functions to get best audio performances and small latency (< 30ms) – available since Windows VISTA
KS	Kernel Streaming or Direct Kernel streaming API allows low latency audio streaming, since Windows XP, but unfortunately not all audio devices provides this interface.
WaveRT	The WaveRT miniport driver is supported in Windows Vista and later Windows operating systems and can offers good audio performances and small latency (comparable to KS).
Direct-X	Direct-X Audio Interface is used by Video Game and some audio software. Latency is usually comparable to MME,
ASIO	Audio Stream Input/Output (ASIO) is a computer sound card driver protocol for digital audio specified by Steinberg, providing a low-latency and high fidelity interface between a software application and a computer's sound card.

Voicemeeter as Audio HUB and Next Gen Audio Engine:

Voicemeeter now includes VBAN features and Voicemeeter Remote API. VBAN allows to send / receive audio stream to / from any computers of a local network. Voicemeeter Remote API allows developing client applications and taking advantage of all Voicemeeter features, inter connection with all audio interfaces type, audio device aggregation, mixing/routing functions, MIDI and VBAN features, to create new powerful audio applications.



Voicemeeter is installed with Macro Buttons Application offering programmable buttons to act on Voicemeeter by simple request script

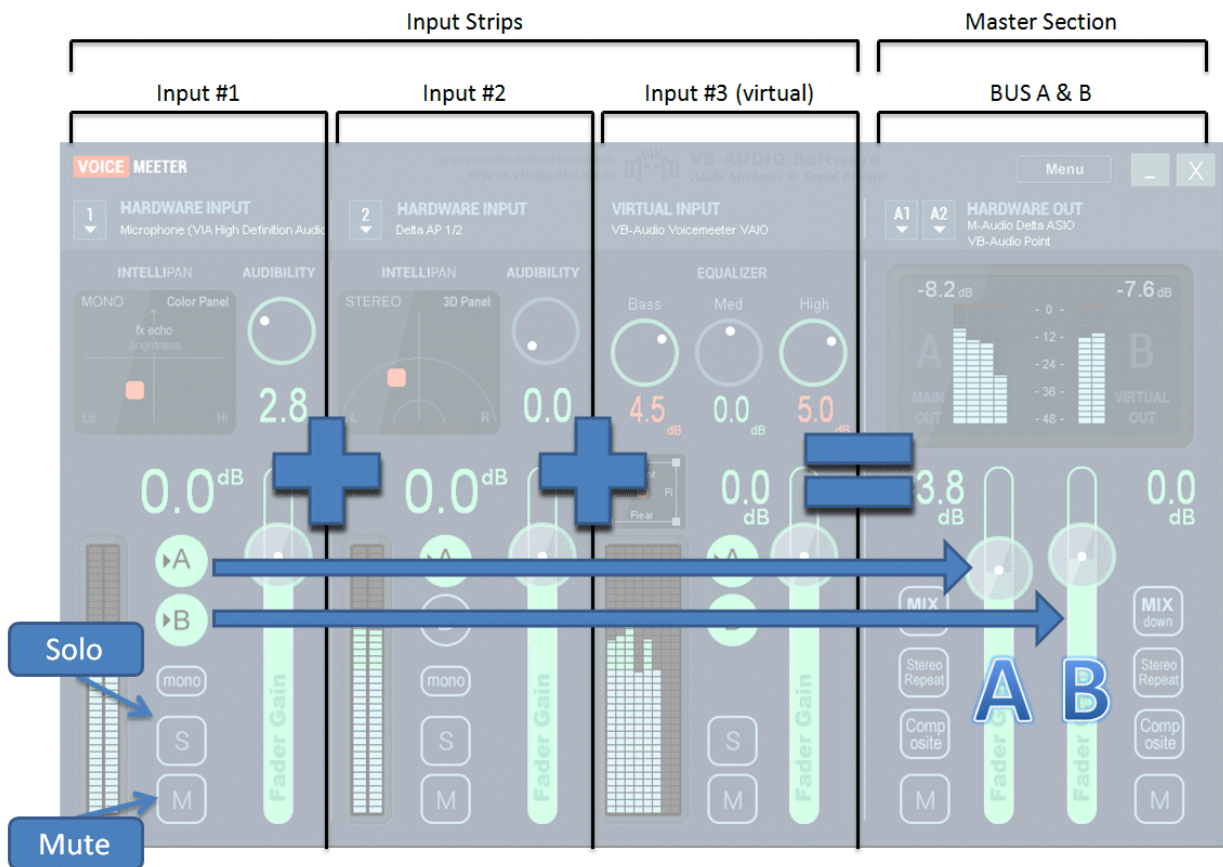
Voicemeeter package also installs VoicemeeterRemote.dll and VoicemeeterRemote64.dll for client application willing to control Voicemeeter by program.

More information and SDK download on our forum:

<https://forum.vb-audio.com/viewtopic.php?f=8&t=346>

Audio Mixing Console Basis:

An audio mixing console is a device able to sum different audio signals (audio sounds) coming from different inputs and route them to different outputs. Mixing desk is usually composed by Strips and Busses. While strips are connected to Inputs, busses are connected to outputs.



This schema above shows how basically Voicemeeter can mix 3 inputs into 2 busses (in BUS A if button A is switched on, on BUS B if button B is switched on). Typically BUS A can be used for monitoring (speakers) and BUS B for VOIP or audio recording applications.

Each strip (each input) can go through different processing before summing to the busses. Voicemeeter includes funny panels to tweak the sound on input #1 and #2 and a regular 3 bands Equalizer on input #3 (the Virtual Input).

Again on usual Mixing Console, we have on each strip a SOLO button (to listen one or several strips only) and a MUTE button to simply mute the signal of the Strip or Bus.


Finally the Fader Gain allows to adjust the sound volume for each Strip and Bus.

Other controls will be explained later, but basically we have described here the outlines of what is an Audio Mixer, generally speaking.

STEP 0: Quick Startup

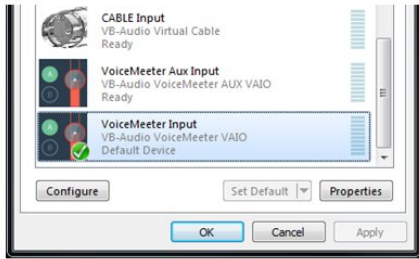
Voicemeeter is a virtual audio mixer intended to replace the Windows Mixer. The quickest way to configure Voicemeeter is to simply use it as default device and connect it to your main playback device. Then all PC sounds will go through Voicemeeter before playing back to your speakers.

1 Install Voicemeeter...




...and REBOOT your computer

2 Configure default Audio Device...



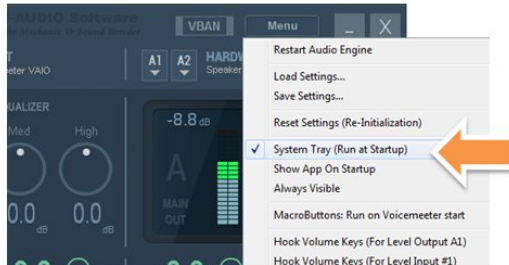
...set Voicemeeter Input as default

3 Launch Voicemeeter Application...



...select Audio Device for output A1

4 Set Voicemeeter In System Tray...



...to replace Windows Audio Mixer

Remarks in case of problem:

Installing Voicemeeter means installing audio driver. In some cases, the system might declare this new driver as default driver (as default playback device for example). Consequently you do not hear anything anymore since you do not run and configure Voicemeeter and your system audio correctly (as it is shown above).

Windows System Audio Configuration is mainly located in Windows Control Panel / Sound Dialog Box. This is where you have to go to check or set what is your default playback device and what is your default recording device (and the Communications option has to be set to "Do Nothing" if you use Voicemeeter with VOIP applications).

In all cases, all is a question of audio configuration and it does not need to remove Voicemeeter in case of problem. Voicemeeter is a simple Application endowed with an audio device driver, if you don't use it, there is no way to get trouble. But if you do, you need to avoid possible audio device conflict and connect everything to Voicemeeter (to prevent other application to use audio playback device already used by Voicemeeter).

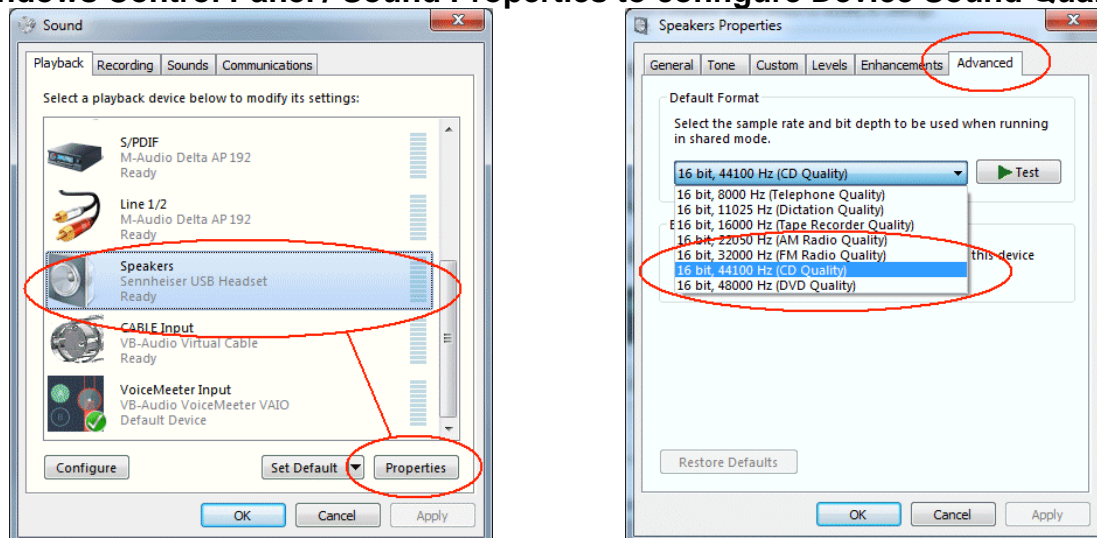
STEP 1: Select main audio Output Device

To make Voicemeeter working, you need at least to select the main output device used for BUS A (A1). We recommend to select first ASIO (if exists) or WDM device type to get best Latency (WDM / KS playback device are used in exclusive mode per default, bypassing the windows mixer and possibly its Volume Control).



The main audio output device (A1) can work in 32 kHz, 44.1 kHz, 48 kHz, 88.2 kHz, 96 kHz, 176.4 kHz or 192 kHz. This will also be the main sample rate of the entire mixing process and Virtual ASIO points (Voicemeeter will support any sample rate on other inputs or outputs points).

Windows Control Panel / Sound Properties to configure Device Sound Quality:



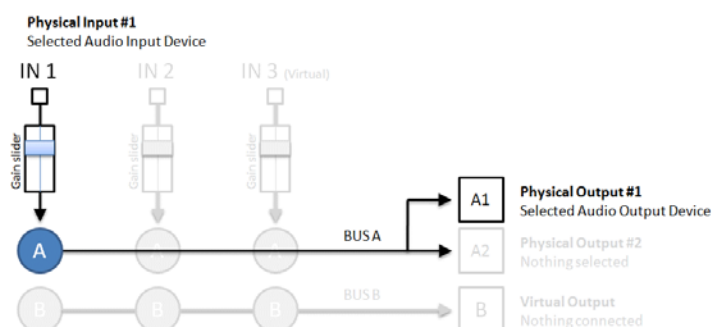
STEP 2: Select an Input Device.

Select audio input device on strip #1, for example your microphone! Then you will be able to listen to your voice in real time.



REM: it's better to select WDM driver (present since Windows VISTA) to get best latency and audio performances. If using MME driver you might have a bigger delay between input and output. Use MME driver only if WDM are not present or does not work correctly with your audio hardware configuration.

KS can be better than WDM in some cases but not all audio device are presenting such audio interface. In the example above, Sennheiser USB Microphone and onboard Audio Device are not present as KS device, but only in WDM and MME.



After Step 2, Voicemeeter is routing signal from selected input (e.g. Microphone) to A1 selected Output. Then you can hear from this audio output the sound coming in input #1. A & B button are switch to send signal in BUS A or/and BUS B.

Monitor Input in Real Time

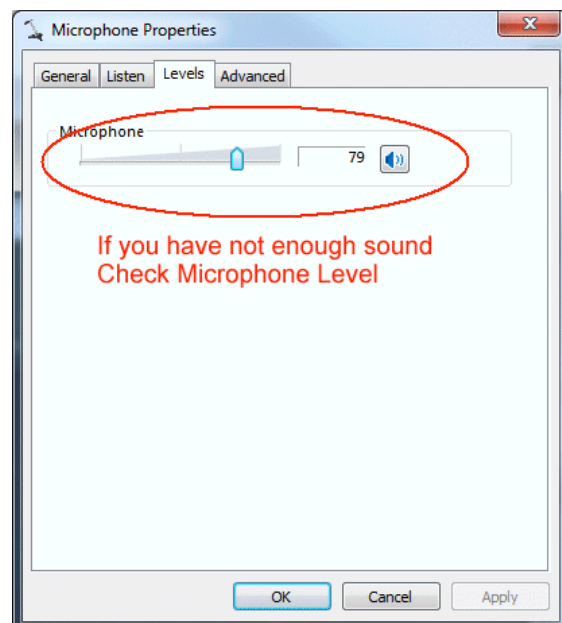
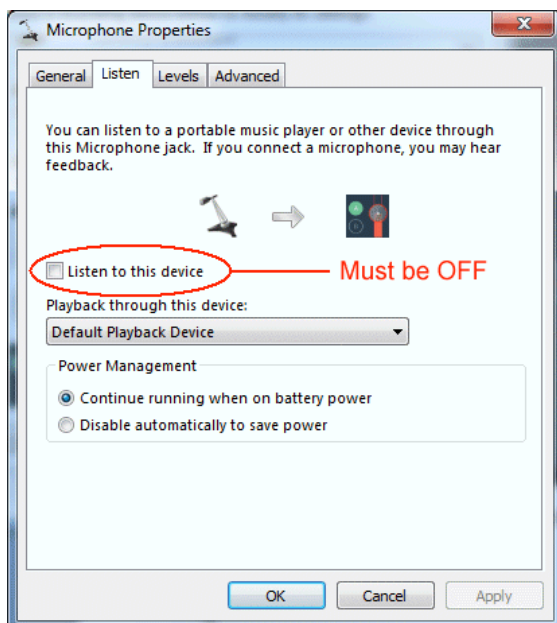
After Step 2, you can hear what is coming in selected input. Blue VU-meter will show you level in real time. Input signal is per default sent to both busses (A & B) and we have routed the BUS A to an output device in Step 1.



WINDOWS AUDIO INPUT CONFIGURATION WARNING

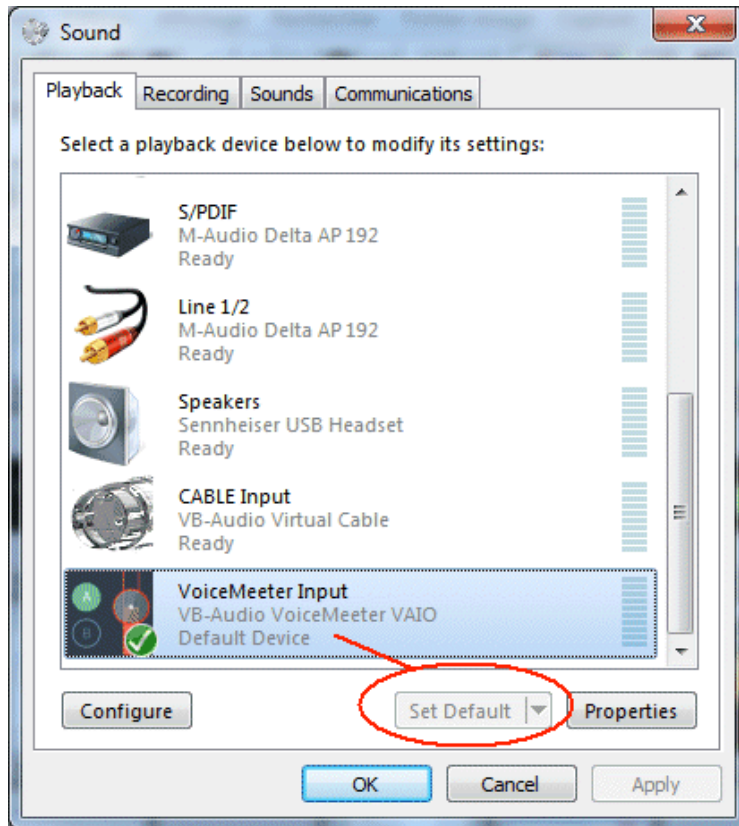
Windows "Listen" option can disturb Voicemeeter routing, so be sure that this option is disabled or makes sense for you.

If your input level is too weak (or too loud), check the level of the selected input device. (Adjust possible pre-amp gain if any).



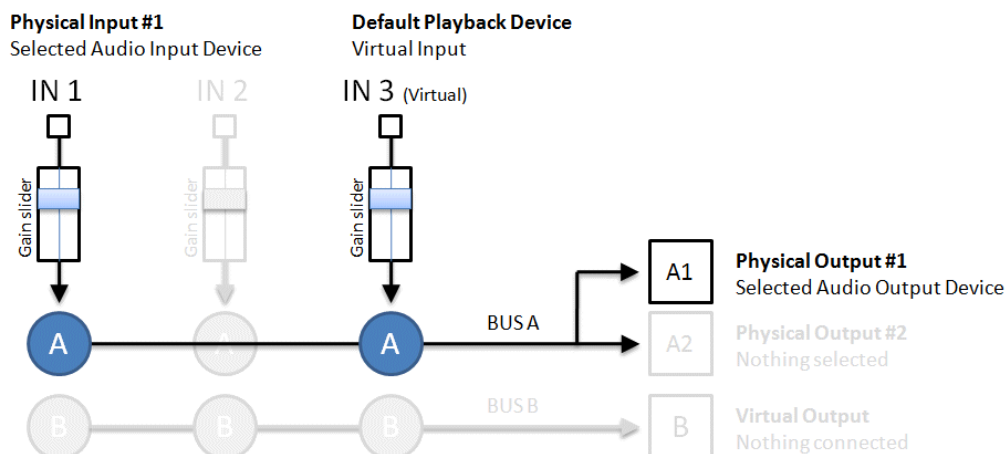
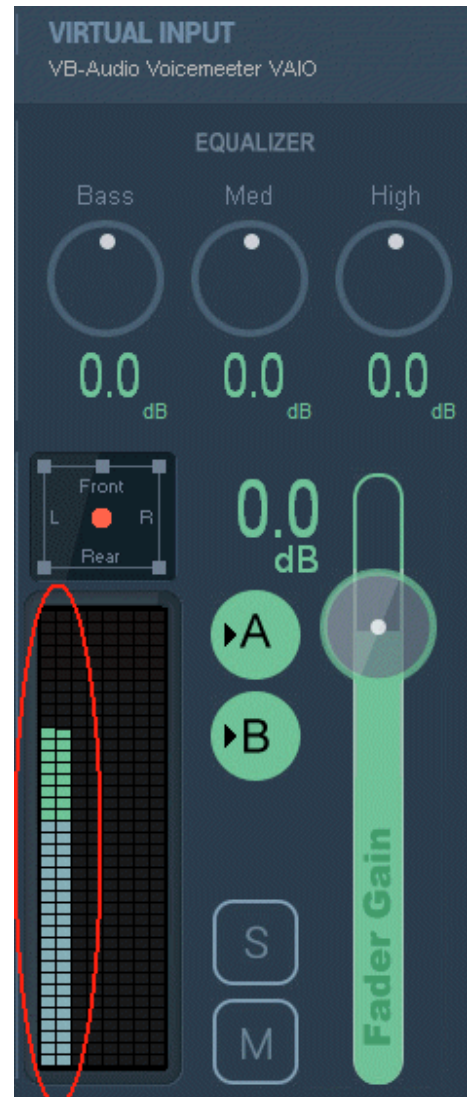
STEP 3: Use Virtual Input as default playback device.

Strip #3 is the virtual input of Voicemeeter. It is shown as regular system audio device and can be used by another application as playback audio device.



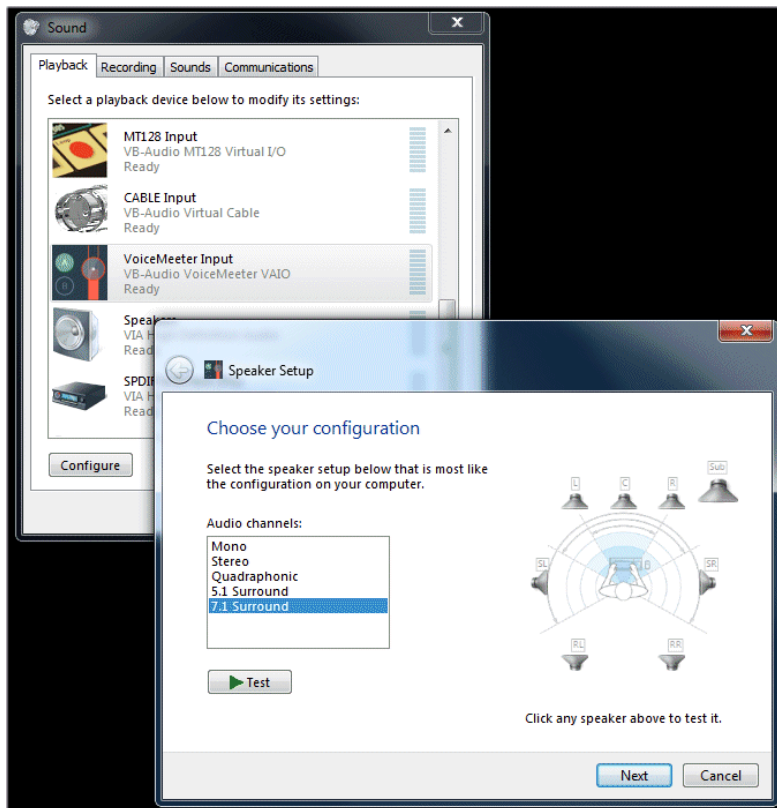
If Voicemeeter Input is set as default audio device, every sound played on the computer will go on the Virtual Input (Strip #3) of Voicemeeter Mixing Console.

You can launch a Media Player or Web Radio and check that you can hear the sound coming from it on the Voicemeeter Virtual Input (IN 3).



Virtual Input is Multi-Channel !

Strip #3, the virtual input of Voicemeeter can manage until 8 channels and get 5.1 audio from DVD for example.



Windows Control Panel /Sound Dialog Box allows configuring Audio device (playback and recording devices).

To get multichannel audio from DVD player you need to configure the Voicemeeter Input in multi-channel (click on configure button on the left bottom after having selected the Voicemeeter input).

When playing DVD you will see 5.1 audio signal incoming on Voicemeeter Virtual Input and also on possible A output level (if your output device supports the same number of channels).

Note that BUS B is also multi-channel but level meter shows always 2 first channels only.



Understand Voicemeeter Virtual I/O in one picture.

Voicemeeter virtual inputs and outputs are simply given by the virtual audio driver playback and recording devices installed on your operating system with Voicemeeter.

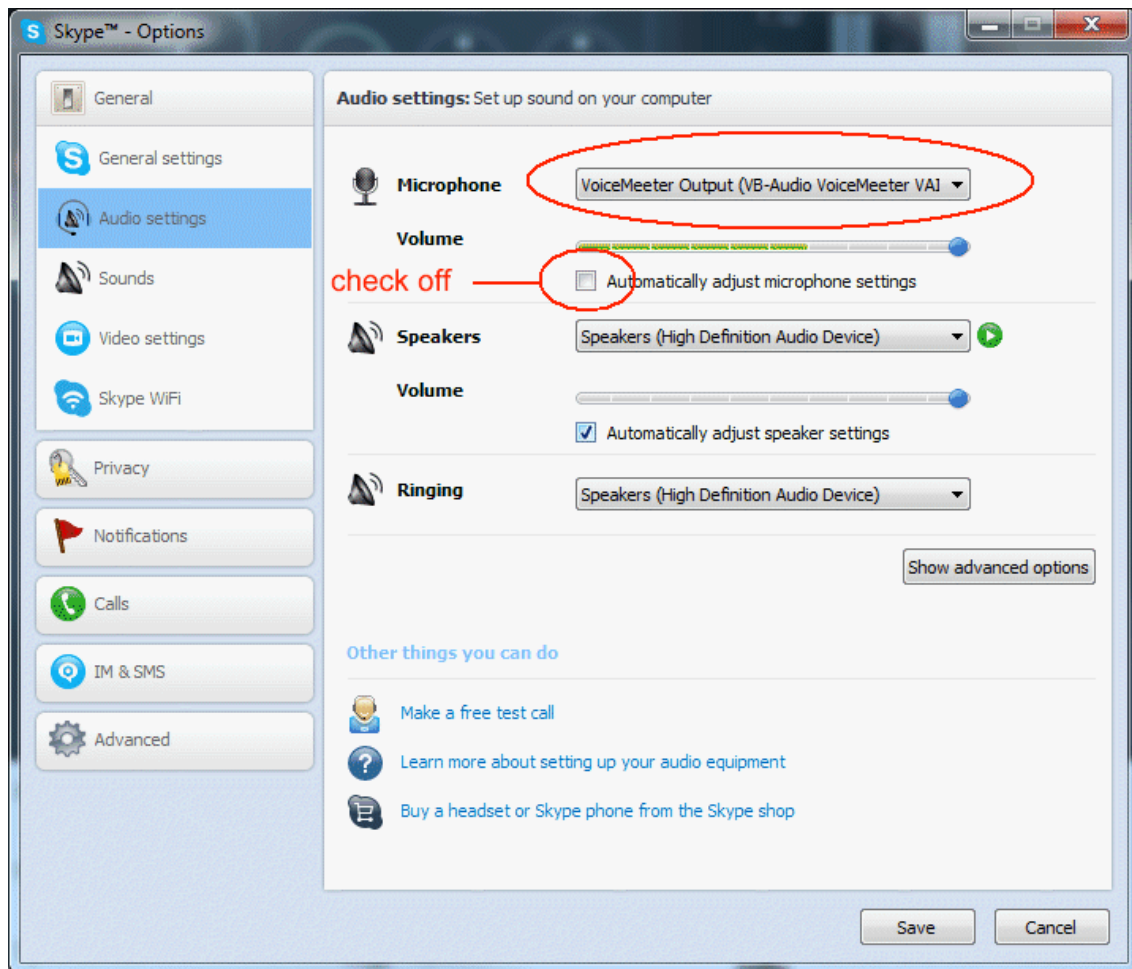
Voicemeeter Virtual I/O

VoiceMeeter playback device is used to get Virtual Input signal

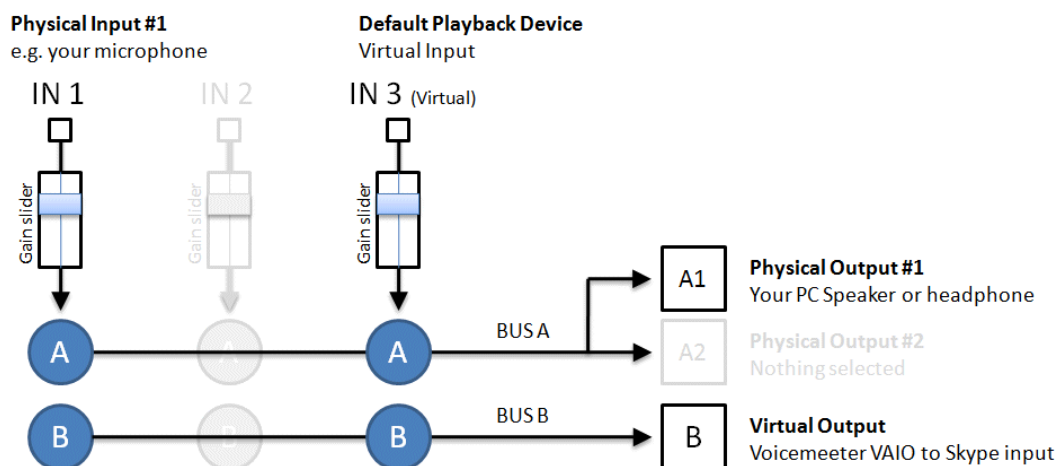
BUS B sends signal to Voicemeeter recording device used as Virtual output

STEP 4: Send the Mix Output to Skype Input.

BUS B, not used yet, can be used to send the mix to virtual output that can be connected to another application (typically a recording or VOIP application).

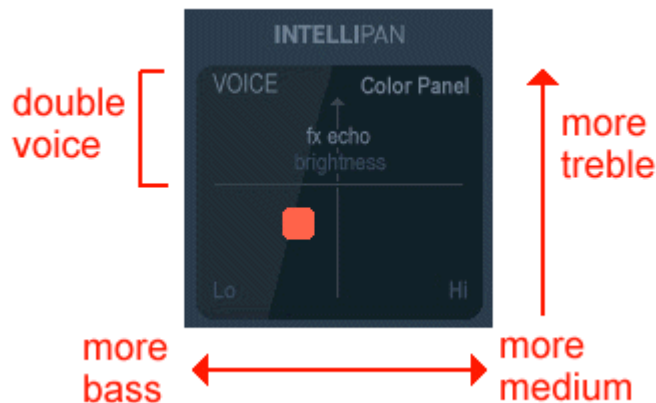


For example in Skype you can select the Voicemeeter Virtual Output as Microphone.



STEP 5: Enjoy new Audio Controls.

While you are talking you can tweak your voice in real time with different controls; 2D Panel, Audibility knobs or simple 3 bands Equalizer...



INTELLIPAN COLOR PANEL

Based on basic equalizer, this panel will allow you to change the color of your voice in a quick way.

It gives a spectral identity to your voice by acting on 3 frequency bands and a tiny reverb on the half top.

INTELLIPAN 3D PANEL: The Binaural effect.

If you right click on panel you can switch to another effect able to position audio sources with a simple room effect (works in stereo only).



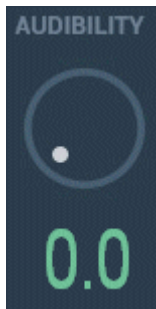
Voicemeeter includes a binaural effect able to give a spatial identity to audio sources and then increase dialog intelligibility (especially when everyone is talking in the same time).

This effect works in stereo, so can be used when recording a stereo stream to improve the sound of the global mix. This also can be used to simply enhance your voice to stereo.

This kind of “psycho acoustic” effect is used a lot since decades in audio production to create a better “audio scene” by positioning some music tracks on different virtual location. Today several FOH live mixing consoles are proposing such advanced Pan Pot, mainly because it allows saving audio energy – while the regular pan pot could completely remove the sound from left or right.

Audibility control & equalizer.

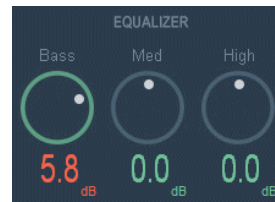
On both hardware input strips, there is an Audibility control and on virtual input there is a regular 3 bands Equalizer.



AUDIBILITY CONTROL

This single knob controls a compressor / gate allowing to boost your voice and manage noisy talk.

It needs to be adjusted according microphone capabilities and sound environment.

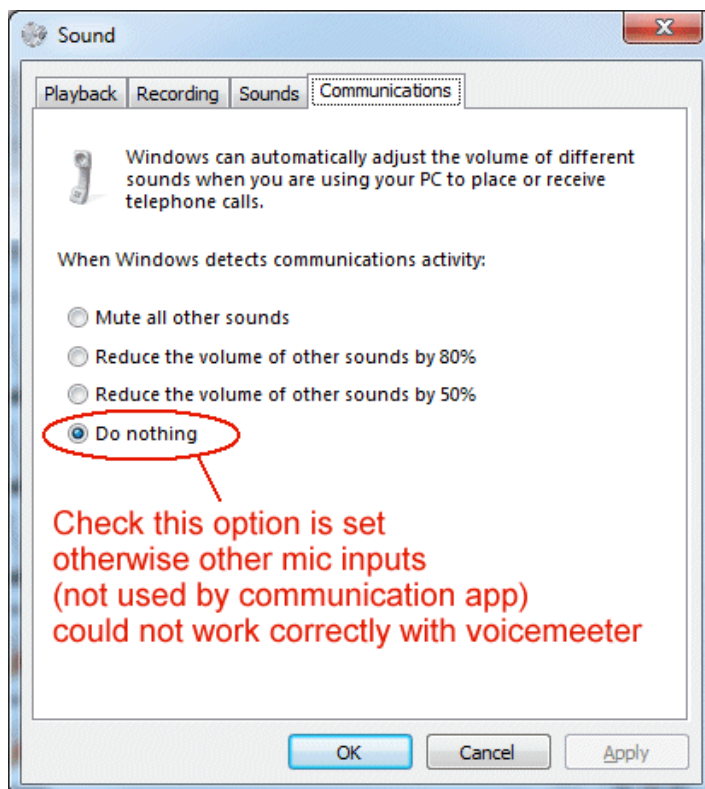


EQUALIZER

3 Bands Equalizer to boost or remove bass, medium and high frequency (treble).

Trick: All controls go back to default value if double click on it !

WINDOWS CONFIGURATION WARNING WITH VOIP



Communication options can mute audio device not used by VOIP Application like our microphone connected to Voicemeeter and no more on Skype.

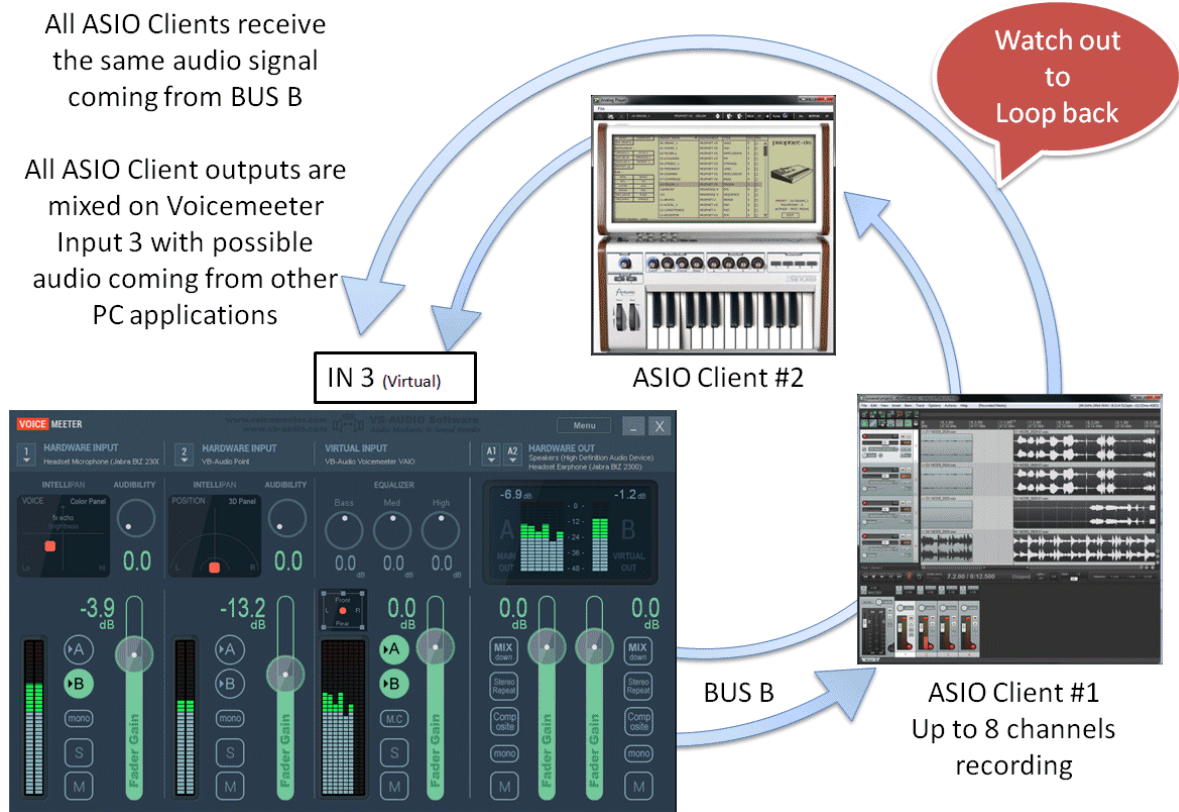
About Built In Microphone:

Built in Microphone can also work not as well as expected: First, speaker in notebook can be captured by built in microphone and generates feedback loop. Secondly built in microphone can have strange behavior according different exotic options present to manage this microphone and communication functions.

We recommend using USB headset or external microphone to also improve sound quality.

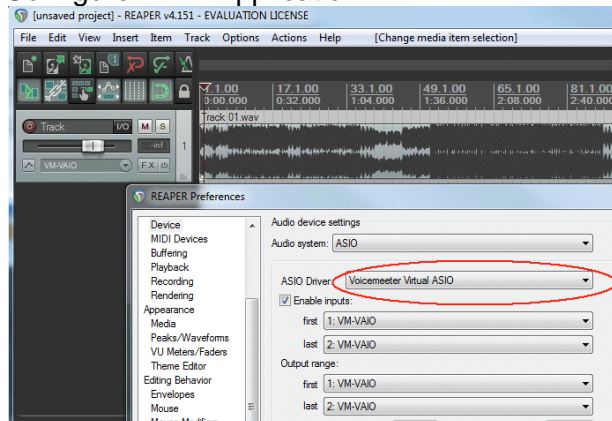
STEP 6: Connecting ASIO Application to Voicemeeter.

Each Voicemeeter Virtual I/O also supports an ASIO interface able to manage 4 client applications. These 4 possible sources get signal from BUS B1 (for Voicemeeter Virtual ASIO) and their outputs are mixed together on the related Virtual Input (with possible PC Sound already coming from usual windows audio interfaces).



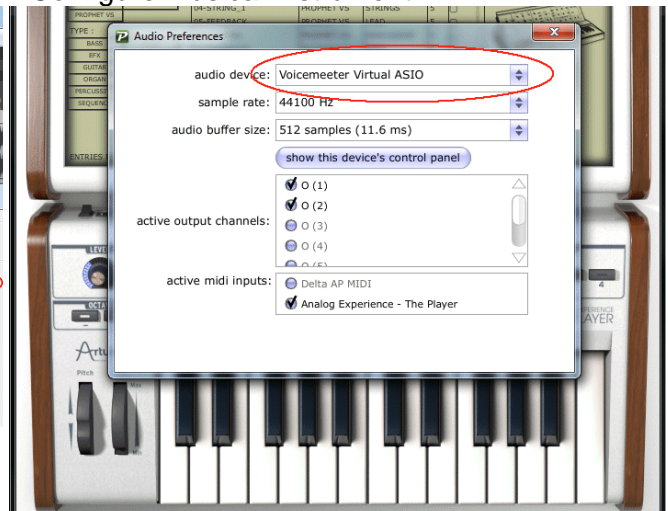
For Recording Applications, take care about the loop back! You might have MUTED outputs or disable monitoring otherwise output signal will come back on input infinitely.

Configure DAW Application



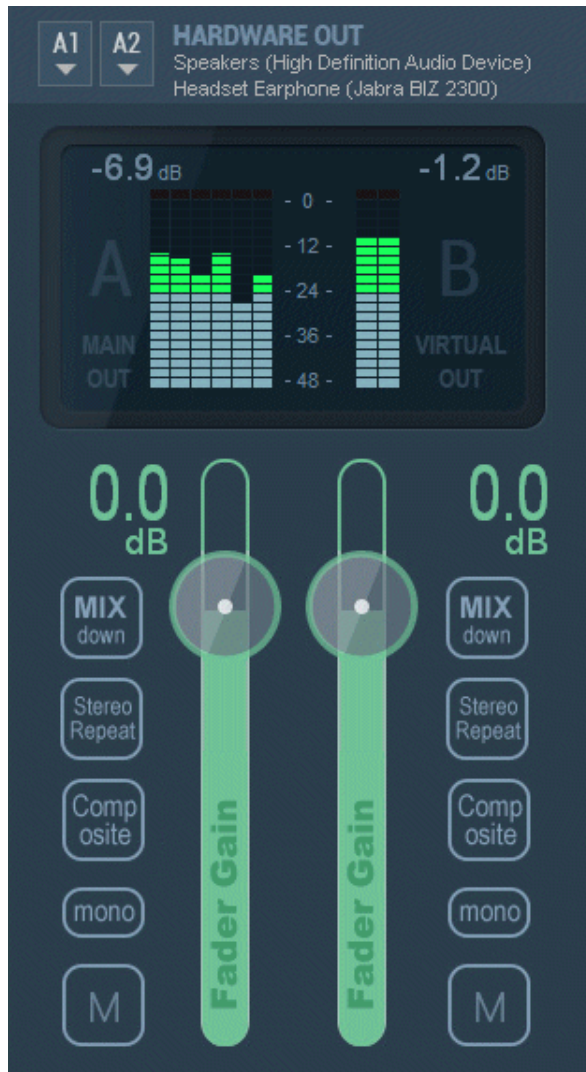
Select Voicemeeter Virtual ASIO device
In any audio applications supporting ASIO

Configure Musical Instrument



Special Routing Options on Output BUS.

Voicemeeter provides 3 additional routing modes for each Busses A & B in order to use the possible 8 output channels in different ways.



MIX DOWN

This button is made to make a stereo mix-down with 5.1 or 7.1 sound coming from DVD player on virtual input (strip #3). Left and right channels, Center, Sub and rear are combined to output on stereo speakers.

STEREO REPEAT

This button is made to use possible 8 outputs channels with a stereo signal. This stereo signal is repeated to channel 3,4 / 5;6 and 7,8.

COMPOSITE

This last button is made for audio post production. The 8 channels are composed in this way:

- ch 1,2 : usual BUS stereo output
- ch 3,4: Voicemeeter input #1 before gain fader
- ch 5,6: Voicemeeter input #2 before gain fader
- ch 7,8: Virtual input channel 1,2 before gain fader

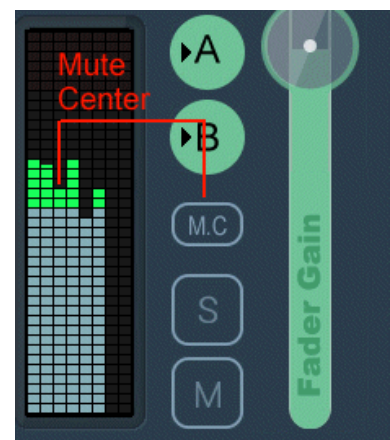
With composite signal, it's possible to record all Voicemeeter Inputs (each in stereo) with a DAW connected to Voicemeeter virtual ASIO for example. It allows recording VOIP interview or conference and making post production process after with the 3 audio stereo tracks.

MONO button simply merges channel 1 & 2 to make mono signal in both channel 1 & 2. Made for single speaker monitoring.

M.C. Means “Mute Center”.

The MUTE CENTER button on Virtual input is made to mute dialog on DVD played in multichannel mode like 5.1 or 7.1.

It allows over dubbing your favorite movies for example...



ASIO Routing Capabilities.

With Voicemeeter 1.0.5.0 / 2.0.3.0 it is possible to route all physical inputs and busses to possibly 64 I/O of the ASIO Driver selected as output A1. It is the optimal way to use Voicemeeter with a professional audio board.

This is possible by selecting no device for physical input / physical bus. Then the system settings dialog box proposes to select input channels / or output channels for bus A2:



In this example:

- Physical Input #2 gets audio from ASIO input channels 3+4
- BUS A1 automatically uses 8 first output channel of the ASIO board
- BUS A2 will send audio to ASIO output channels 63+64

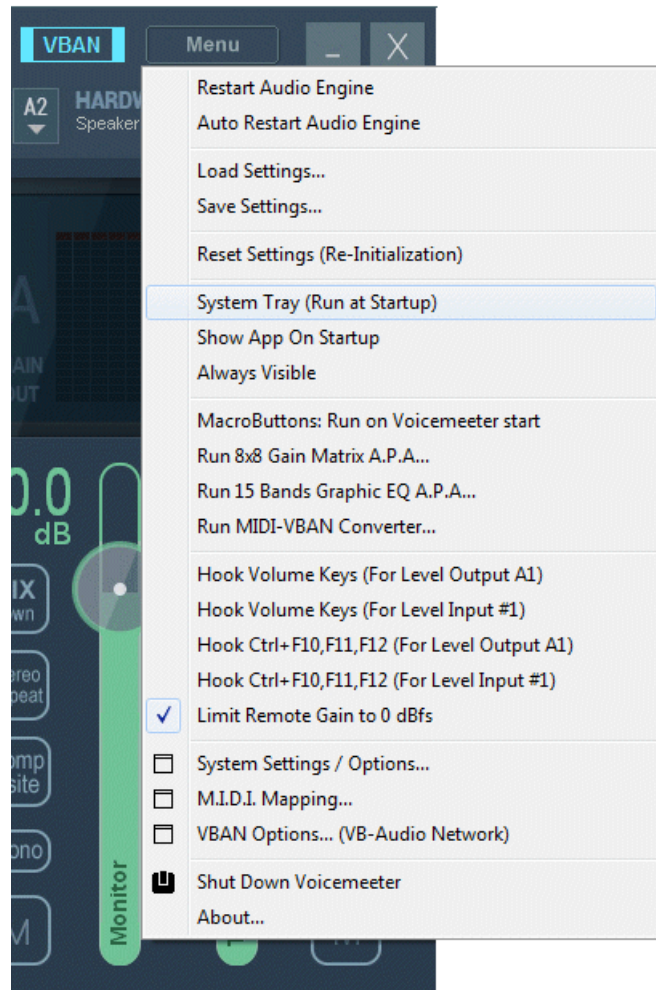
WARNING: BUS outputs are copied in the ASIO output channels in logical order A1, A2, A3. It means that A1 channels (1 to 8) could be replaced by A2 channels (if overlapping). For example if A2 is routed to channel 1+2, it will overwrite output A1 channel 1+2 (you won't hear the BUS A1 anymore, since it is replaced by A2).

Menu for Extra functions

By clicking on “menu” area, user gets access to different functions:

The additional menu is there to provide additional functions:

- To Restart Audio Engine.
- To Automatically Restart Audio Engine if the Output A1 is disconnected / reconnected.
- To load or save current settings in a given filename.
- To reset Voicemeeter complete settings
- To set Voicemeeter in system tray (in this case it will run on system startup).
- To decide if Voicemeeter is shown on Windows startup.
- To set Voicemeeter as always visible (window on top).
- To Run Macro Button on Voicemeeter launch. To run other applications installed with Voicemeeter.
- To hook key to control volumes on Voicemeeter with keyboard (Output A1 or Strip #1 level).
- To open different configuration dialog boxes (System settings, M.I.D.I. mapping, VBAN Dialog Box).
- To shutdown the application.
- The About Box will show you some link to get more information and donate on our webshop.
- System Settings Dialog Box allows to configure parameters related to audio device management..
- M.I.D.I. Mapping Dialog Box allows to connect a M.I.D.I. Remote surface to Voicemeeter.
- VBAN Dialog Box will help you to configure the VB-Audio Network functions, to send/receive audio stream to/from any computer of your local network.

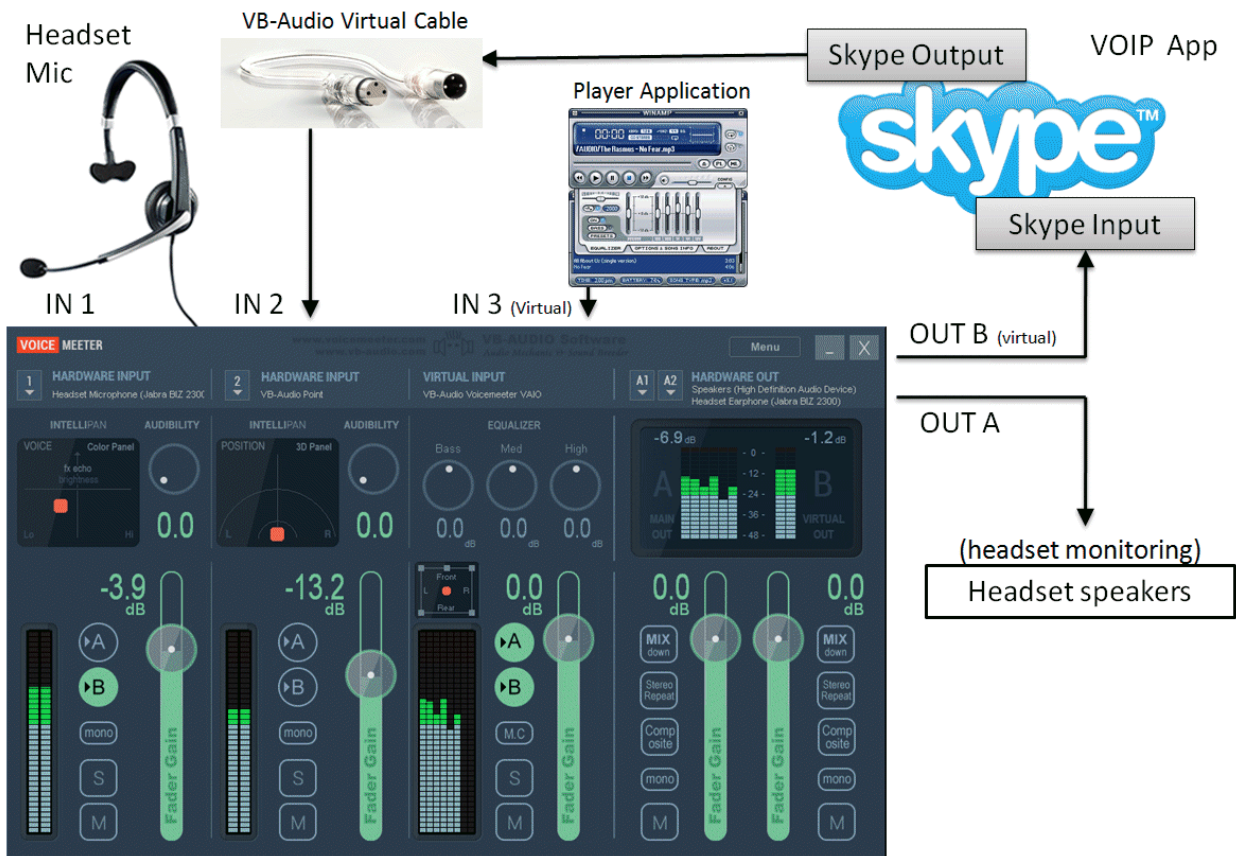


CASE STUDY #1

How to talk and send music in the same time on Skype?

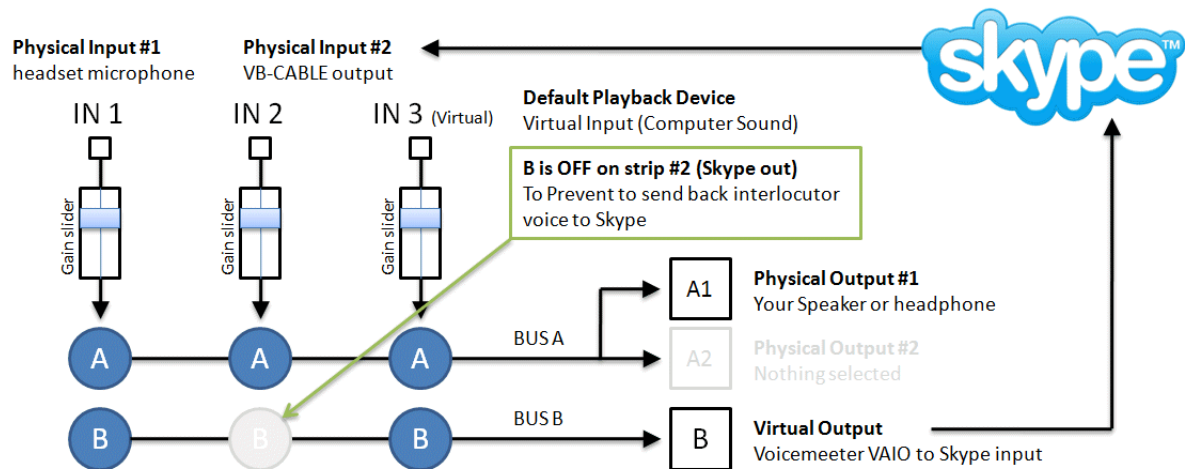
CASE STUDY #1

How to talk and send music in the same time on Skype or Google Voice? To make it, we will use a configuration very closed to our current setup after Step 5 of this document.



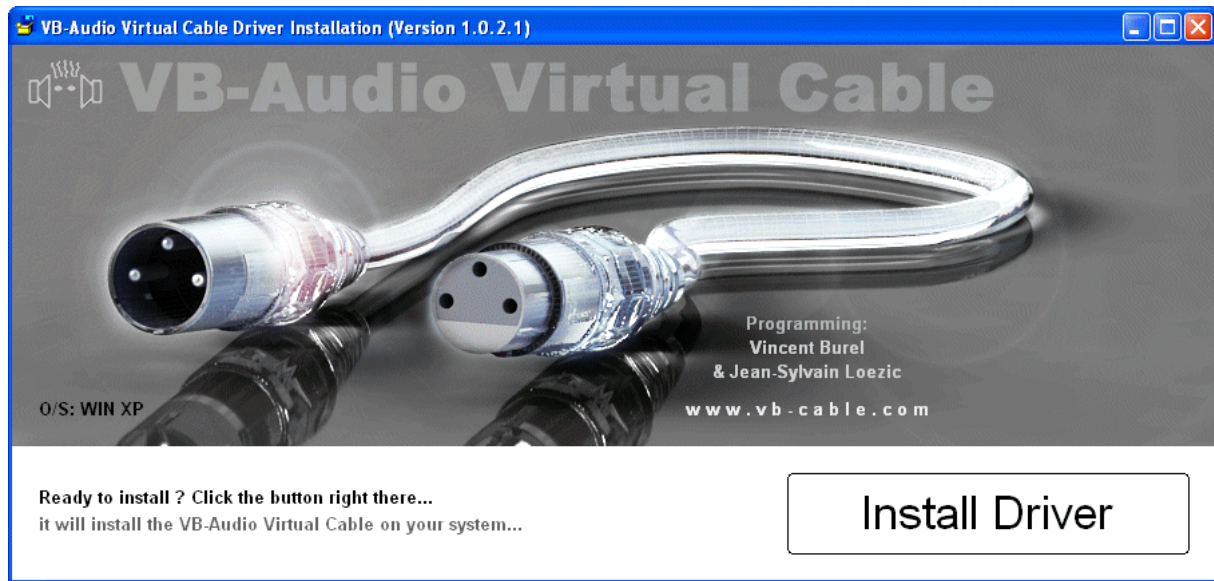
For this configuration, we need to install VB-Audio Virtual Cable, it makes the things more comfortable since we have control on the 3 sounds: our voice (IN 1), Skype output (IN2) and computer sound (IN 3).

VB-Audio Virtual cable will be used to connect the Skype output to Voicemeeter Input #2.

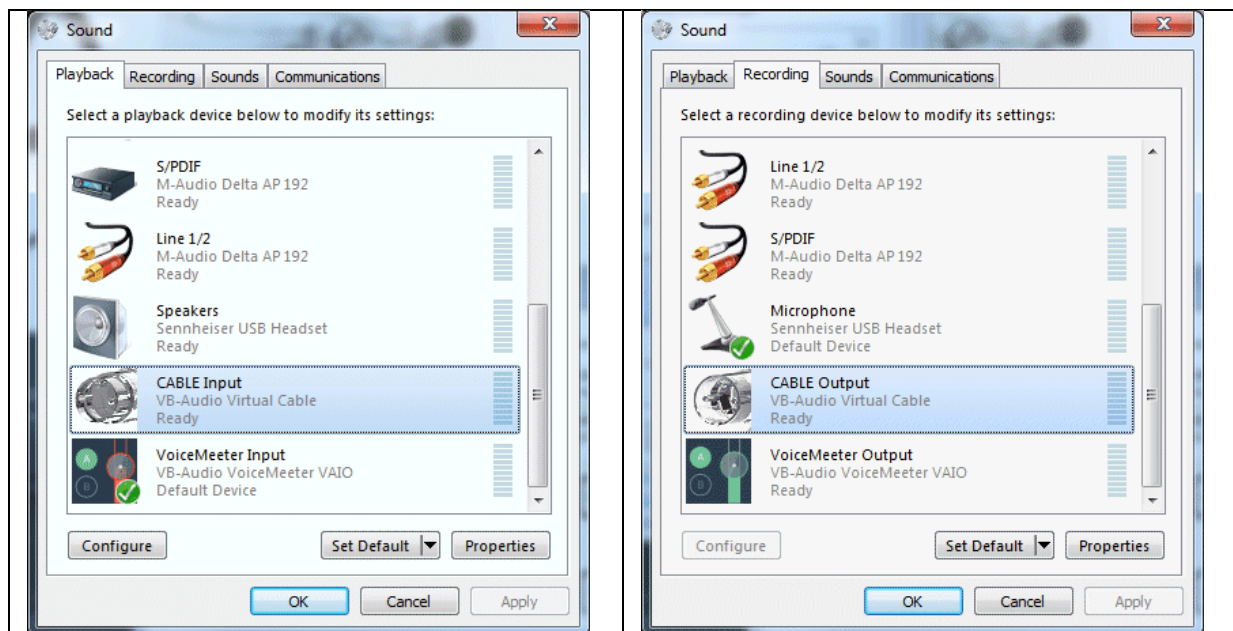


CASE STUDY #1 – STEP 1

Install VB-AUDIO Virtual Cable: download it on www.vb-cable.com

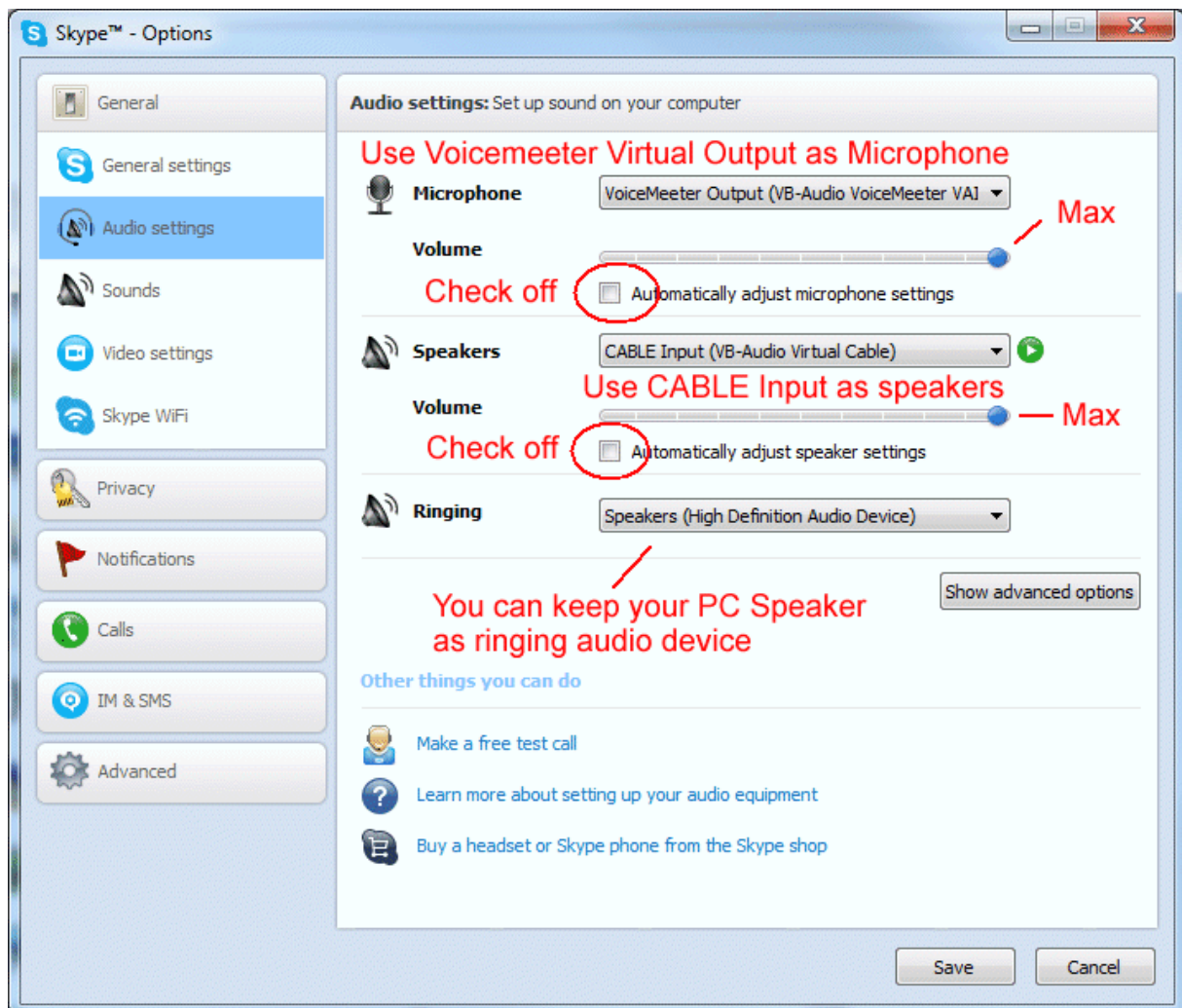


After Installation, you have a new audio installed driver called CABLE Input (as playback device) and CABLE Output (as recording device). These are the two ends of the cable, and like every cable, all sounds sent to cable input will go on cable output.



CASE STUDY #1 – STEP 2

Configure Skype to receive audio from Voicemeeter B Output (Virtual Output) and send audio to CABLE (newly installed).



Note it's important to check off "Automatic gain control" because it's now Voicemeeter mixing console which takes care about levels. Also you can set level to max for the same reason.

In this above configuration, Skype will still continue to use your PC speaker to ring you, but for communication, you will need to launch Voicemeeter to make it work of course (see menu to set Voicemeeter in system tray and run on Windows startup)..

CASE STUDY #1 – STEP 3

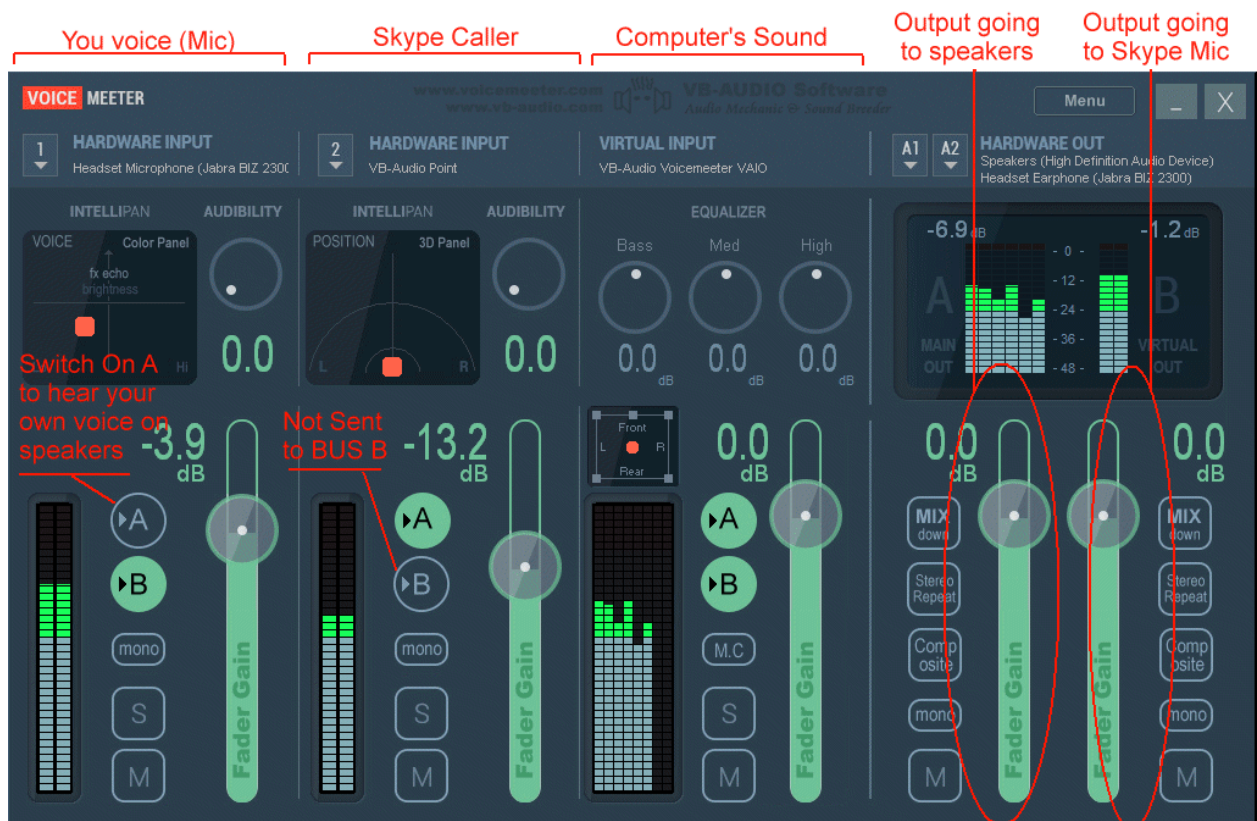
Configure Voicemeeter to connect it to skype through the VB-CABLE.

Already done in last document steps:

- Select A1 Output device: your headphone/headset speakers.
- Select Input device IN1: your headset microphone.
- Set Voicemeeter Virtual Input as default playback device (to get all computer sound in IN3 – all computer sound played on default playback device of course).

To be done to get skype output:

- Select Input device IN2: CABLE output (where Skype output is routed to).



With A / B Switches, you can decide what sound you send to monitor (speaker) and what you send to Skype (through Bus B).

Basically you will send to your headphone everything but your voice (except if you want to also hear your own voice in speaker). That's why A is off on strip #1.

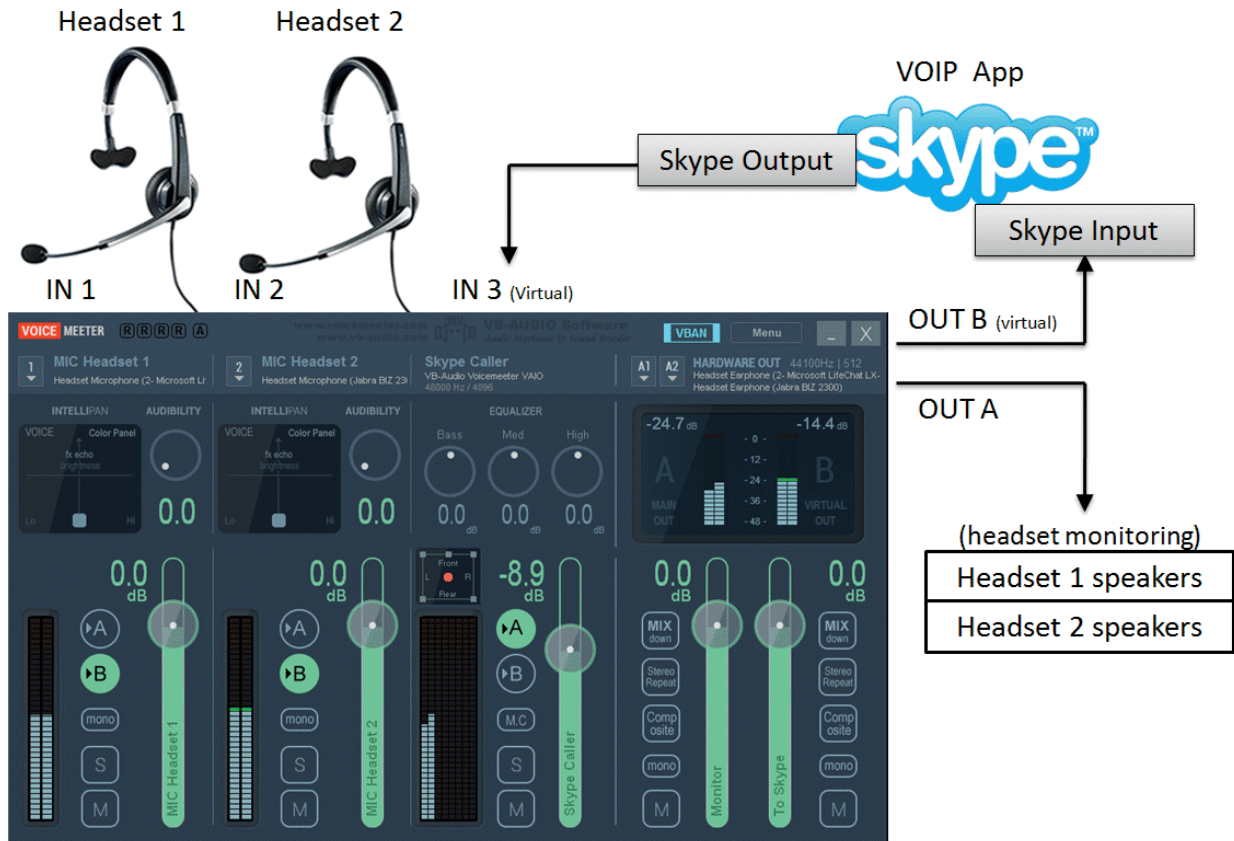
Regarding Skype, you will want to send everything (your voice and music) but skype sound (Skype caller coming in input 2). That's why B is off on strip #2. Otherwise we could have a loop back and the caller could hear his own voice again and again and again...

CASE STUDY #2

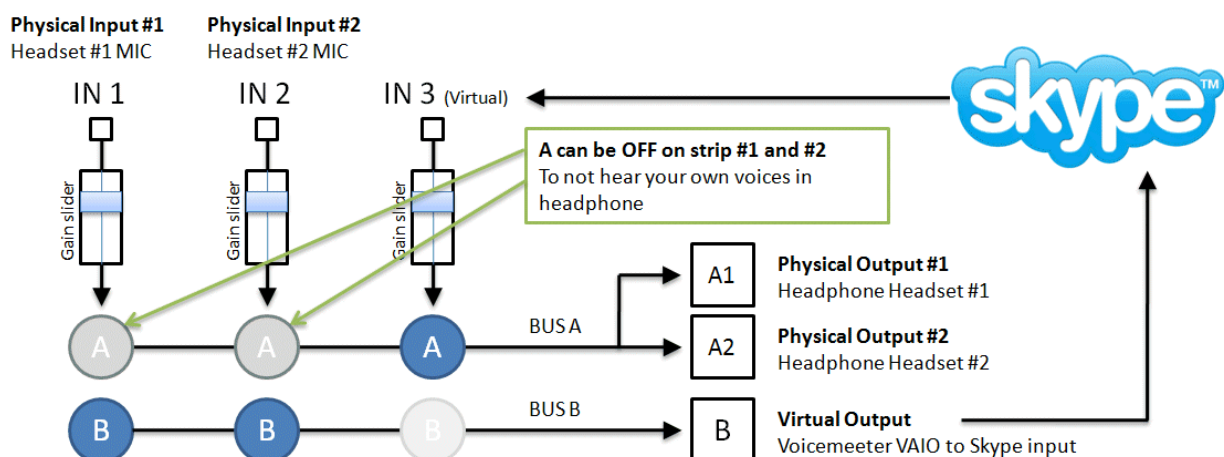
How to manage 2 headsets on Skype?

CASE STUDY #2

How to manage 2 Headsets in the same time on Skype or Google Voice? To manage 2 headsets is maybe simpler than the previous case, since we don't need additional VB-Cable.



To manage 2 Headset, we need to manage 2 different microphones, but also 2 different headphones. That's why Voicemeeter allows selecting 2 physical output devices on BUS A outputs (A1 and A2). Then both headset speakers will give the sound coming from Skype (IN3).



B button on strip #3 must be OFF to avoid Skype feedback (otherwise audio coming from Skype on strip #3 would be sent again to Skype through Voicemeeter Virtual Output).

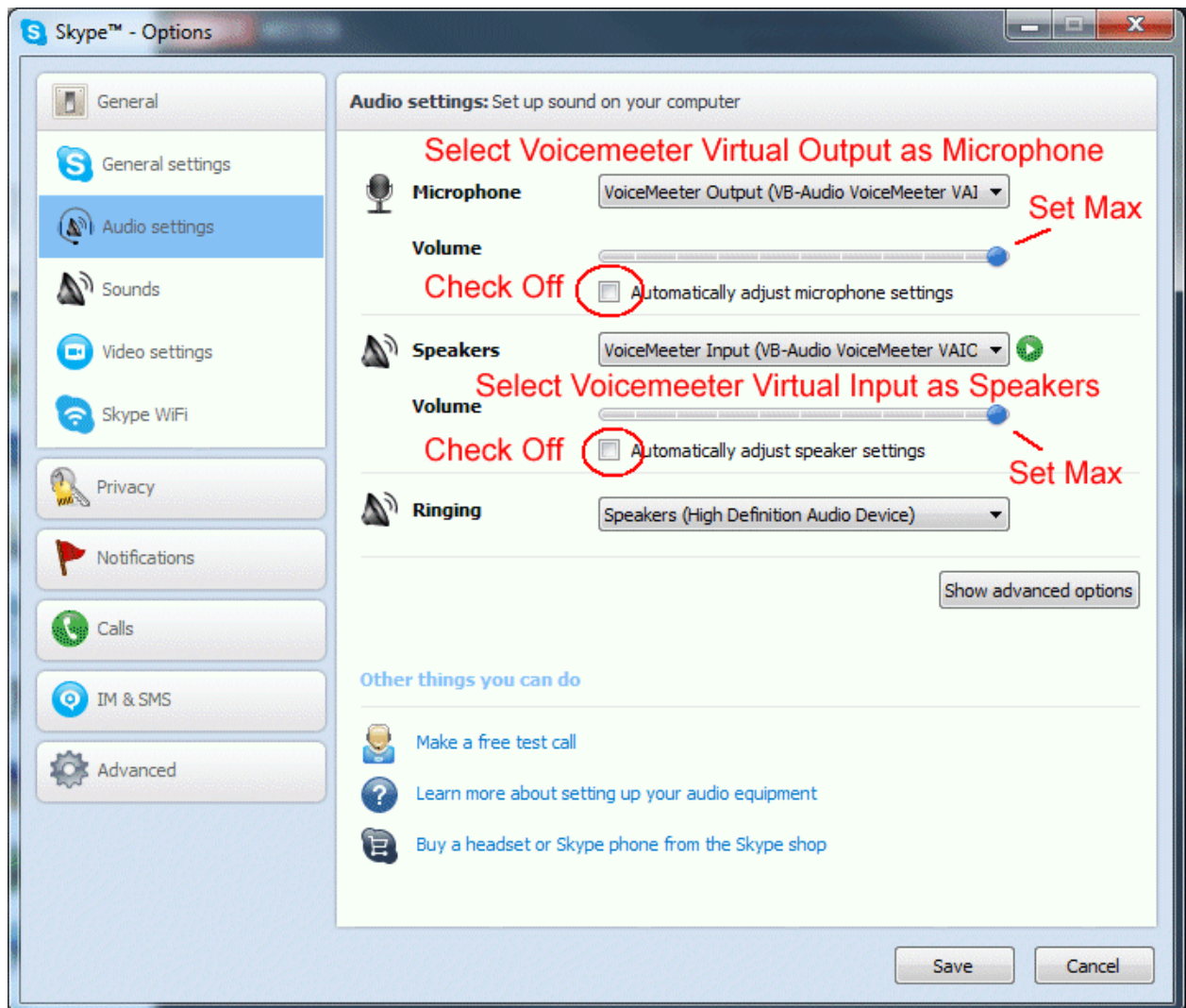
CASE STUDY #2 – STEP 1

Configure Voicemeeter to manage 2 headsets

It just means to select audio output A1 and A2 for your headphone 1 and 2. Then Select Input Device 1 for your microphone headset #1 and Device 2 for your microphone headset #2.

CASE STUDY #2 – STEP 2

Configure Skype Audio Settings to use both Voicemeeter virtual audio i/o :



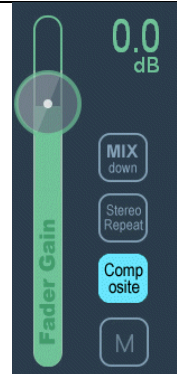
Do not forget to click on SAVE button to validate your settings. Now you are ready to use VOIP application with 2 Headsets.

CASE STUDY #3

How to record Conference-Call
in 8 tracks for post production?

CASE STUDY #3

Since Voicemeeter is providing ASIO interface, we can connect audio pro DAW to Voicemeeter, not only to playback into Voicemeeter Virtual input, but also to record the Voicemeeter Virtual Output (which is composed of 8 channels).



If we connect for example REAPER to the Voicemeeter Virtual ASIO point, we can manage 8 channels recording. And the COMPOSITE buttons is there to feed these 8 channels with the 3 Voicemeeter inputs before fader (pre fader).

Of course channel 1,2 remain unchanged compared to other modes, but the 6 other channels are replaced by Voicemeeter inputs. This allows for example to make multi track recording of interview made on Skype or Google Voice and record everything on a pro audio DAW to make post production after.

Be careful to disable input monitoring in your recorder application to avoid feedback loop (prevent signal to go again into Voicemeeter virtual input).



REAPER recording 4 stereo tracks coming from Voicemeeter virtual ASIO output.

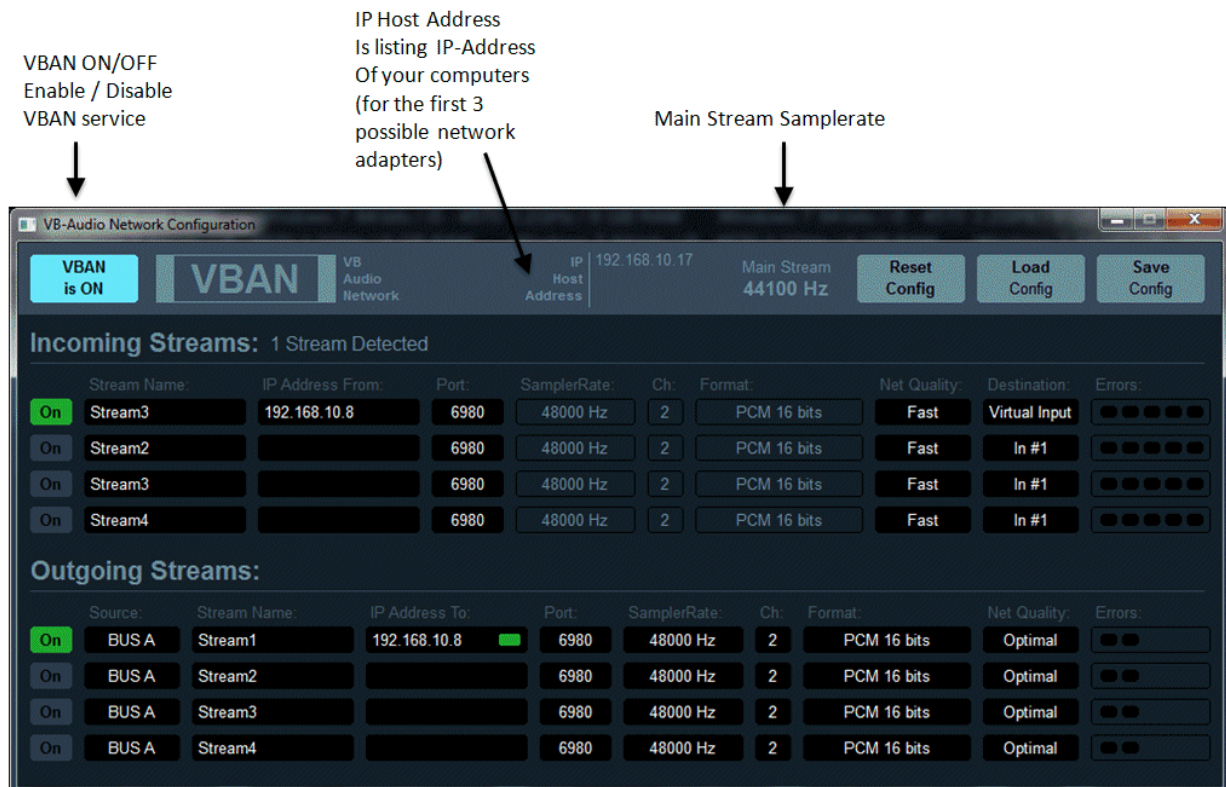
VBAN

VB-Audio Network

VBAN: VB-Audio Network

VBAN Protocol, based on a simple UDP protocol, has been designed for real-time transport of digital audio stream in IP-based network environments. It provides an easy ways to send / receive audio to / from any computers on a **local network**.

Click on VBAN icon to open the configuration dialog box below and configure your incoming stream (the audio stream from other computer you want to hear) and your outgoing stream (the audio stream you want to send to other computers).



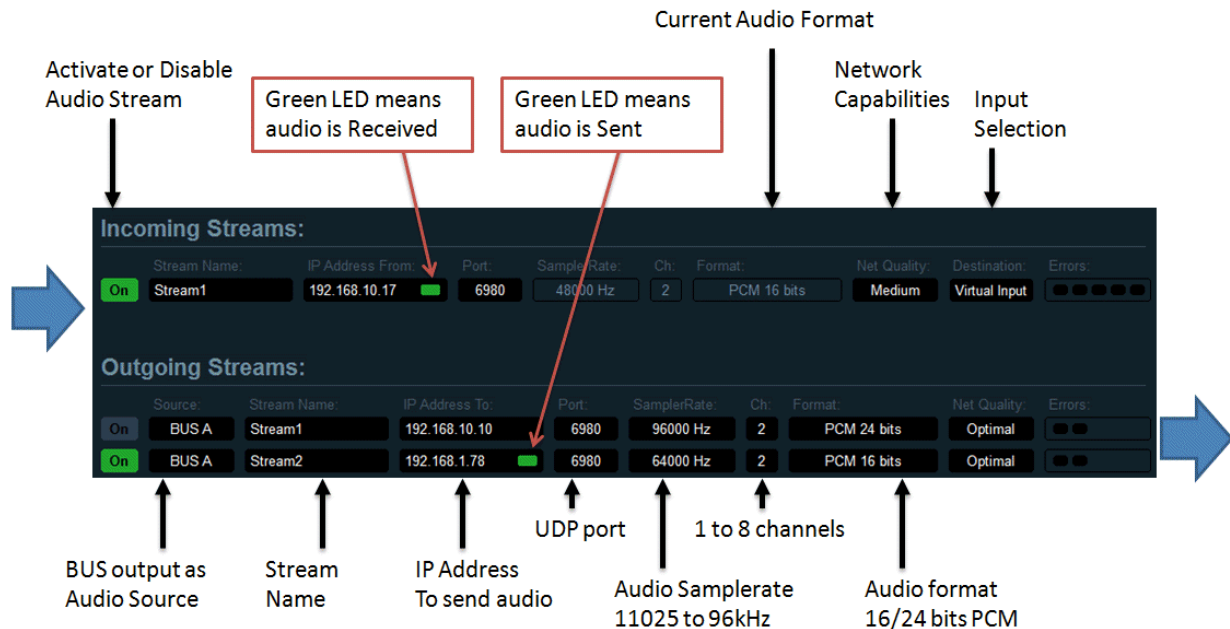
With Voicemeeter, it is possible to listen to 4x streams (on any input) and send 4 streams to different computers or mobile device (audio source is given by BUS A or BUS B Source).

A VBAN Stream in Voicemeeter can be set in any standard samplerate from 11025 Hz to 96 kHz in 16 or 24 bits resolution with 1 to 8 channels (mono sound to 7.1). It allows transporting high quality audio in native PCM format on your local network.

It is possible to broadcast an audio stream by using the 255 IP Address. For example giving 192.168.1.255 as destination address will send stream to all computers of the network 192.168.1.xxx (REM: such broadcast UDP stream could not pass through router or WIFI Access Point – usually forbidding UDP broadcast)

Configure VBAN Audio Stream

We can configure VBAN stream just by clicking on black fields in the dialog box. An edit box or popup menu will appear to let you define parameters. UDP port used by VBAN is 6980 per default. An Audio Stream is defined by its NAME, IP-ADDRESS FROM (and UDP port). To receive an audio stream these 3 parameters must be the same on receiver.



Send audio stream to other computer:

To select an outgoing stream, define stream name and destination IP-address of the target computer of your local network and switch it ON. The audio stream being sent will come from the selected source (BUS A or BUS B)

Receive audio stream from other computer:

On incoming stream, RIGHT CLICK on "Stream name" area or "IP Address from" area to get the list of current detected incoming stream and select it directly. Otherwise it is possible to edit Stream Name and IP-Address manually. Switch ON the Stream to receive audio on virtual input strip.

NETWORK QUALITY: This parameter, more useful on incoming stream, allows tweaking the audio stream stability in case of bad or busy network. FAST means the network is able to transport audio without delay or error, SLOW is made for busy network, where audio packet can be delayed or lost more often than usual.

ERROR LED: shown on the right are blinking red in case of reception errors:

- 1- Overload: we received too much packets (audio stream comes too fast)
- 2- Corrupt: we received corrupted packets
- 3- Disorder: we received older packets
- 4- Missing: we missed packets (lost)
- 5- Underrun: we received not enough packets (audio stream comes too slow)

Additional VBAN-MIDI or Command stream.

Incoming streams section offers 2 other specific VBAN Streams to receive MIDI or TEXT. Opposite to Audio Stream, MIDI or TXT streams do not need an "IP-Address From" and can manage any message coming from anywhere. This is typically done to allow remoting Voicemeeter from several points in the same time. Note the MIDI Stream also feeds Voicemeeter MIDI Mapping (that is also received by MacroButton).

	Stream Name:	IP Address From:	Port:	SamplerRate:	Ch:	Format:	Net Quality:	Destination:	Errors:
On	Stream1	192.168.10.17	6980	48000 Hz	2	PCM 16 bits	Fast	Virtual In #1	
On	Stream1	192.168.10.11	6980	48000 Hz	2	PCM 16 bits	Fast	In #3	
On	Stream3	192.168.10.10	6980	48000 Hz	2	PCM 16 bits	Fast	In #3	
On	Room 3	192.168.10.52	6980	48000 Hz	2	PCM 16 bits	Fast	In #2	
On	Stream5	192.168.10.53	6980	48000 Hz	2	PCM 16 bits	Fast	In #2	
On	Stream6		6980	48000 Hz	2	PCM 16 bits	Fast	In #1	
On	Stream7		6980	48000 Hz	2	PCM 16 bits	Fast	In #1	
On	Stream8		6980	48000 Hz	2	PCM 16 bits	Fast	In #1	
On	MIDI1		6980	115200 bps Serial		90 00 00		MidiMap	
On	Command1		6980	256000 bps UTF8		strip(0).mute=1,strip(1).mute=1,strip(2).mute=		Remoting	

No address = any Last received TXT message Last received MIDI message

VBAN MIDI messages and TEXT commands can be sent by MacroButton (see next pages).

VBAN MIDI messages can also be sent by VBAN2MIDI application (installed with Voicemeeter).

VBAN2MIDI application.

This simple standalone application allow converting a physical MIDI input into a VBAN outgoing stream (LEFT Side) and the opposite: from a VBAN MIDI Stream, send incoming data to a physical MIDI output (Right Side).

VBAN2MIDI-Converter (Version 1.002)

M.I.D.I. Input Device: Analog Experience - The Player M.I.D.I. Output Device: Delta AP MIDI

MIDI input → VBAN → VBAN → MIDI output

OUTGOING VBAN STREAM INCOMING VBAN STREAM

Stream Name:	IP-Address To:	UDP Port:	Stream Name:	IP-Address From:
On MIDI1	192.168.10.255	6980	On MIDI1	any

80 35 17 MTC: 00 : 00 : 00 . 00 80 35 17

MACRO-Buttons

Voicemeeter Remote API

MACRO Buttons

Delivered and installed with Voicemeeter, MACRO BUTTONS Application allows creating programmable buttons to control Voicemeeter parameters. It is displaying several (4 to 80) push or 2 positions buttons with a title / sub title and the possibility to assign a keyboard shortcut / mouse / Game pad / MIDI message or even Audio Trigger.



MACRO Button is an independent application using Voicemeeter Remote API. It is also demonstrating the capability to develop a client application to control Voicemeeter by program. This is made to allow creating any new applications, with custom graphic user interface, to make automation, to make VCA or Auto Ducking process or even to process audio directly inside Voicemeeter thanks to the different possible audio interfaces and API's.

MACRO BUTTON application has been designed for Voicemeeter users, to make custom action, or group of actions on the different Voicemeeter controls and parameters:

- To mute a strip or a bus.
- To change the gain on one or several strips / busses.
- To change or toggle BUS assignments on one or several strips.

It is also possible to combine different requests to create a complex action on Voicemeeter. For example a PUSH TO TALK or AUTO DUCKING function could set the MUSIC Gain to -10db and mute other talker in the same time.

- To change voices color/audibility to make special announces.
- To Restart Audio Engine.
- To save or load a complete configuration file.

On Voicemeeter BANANA version, it is also possible to:

- Make voice FX by changing Modulation and Color Panel.
- Launch Sound by the integrated audio player
- Make special correction with the bus parametric EQ
- Remote all VBAN functions

Finally MACRO Buttons also provides system functions to:

- Send Keyboard Event to System Queue (to remote other applications).
- Execute any program (with possibly a command line).
- Send M.I.D.I. Message to 2x possible devices.
- Send VBAN-MIDI / VBAN TXT requests.

Configuring MACRO Buttons

Right click on the button to open the dialog box below and configure your button. It is basically possible to define 3 scripts of request(s); one for initial condition(s) (sent on startup), one sent when the button is pushed and one when the button is released.

Button Configuration...

Button Name: PTT

Button Sub Name: Push To Talk

Logical ID: 0

Button Type: 2 Positions

Button Color: Default

Keyboard Shortcut: SHIFT+ 0 key

Request For Initial State: //strip[0].mute = 1.0f

Request for Button ON / Trigger IN: strip[0].mute = 0.0f
strip[4].gain = -20.0

Request for Button OFF / Trigger OUT: strip[0].mute = 1.0f
strip[4].gain = 0.0

M.I.D.I. Implementation: ☐ Learn (From MIDI mapping device) Reset

XINPUT: ☒ Enable Ctrl: any GamePad Button: START

HID Device Button: -none- 1

TRIGGER: ☐ Enable

Strip	In	Out	Hold	Level Option
in #1	-20.0	-20.0	1000	<input type="checkbox"/> After Mute

Ok Cancel

Button type can be PUSH or 2 POSITIONS. Keyboard shortcut dropdown list proposes different Key and Mouse buttons combinations. Button color proposes 9 different buttons backgrounds.

M.I.D.I. Implementation allows assigning one M.I.D.I. Event to the button as well. The M.I.D.I. codes are coming from the M.I.D.I. device selected in the Voicemeter M.I.D.I. Mapping Dialog Box and possibly incoming VBAudio MIDI stream. Check LEARN box to get message from M.I.D.I. Device. Click on RESET to remove the M.I.D.I. Code.

TRIGGER will allow handling the button according 2 thresholds on a selected input strip level. IN threshold (green cursor) will push the button when level goes over... OUT threshold (red cursor) will release the button when the level goes under. HOLD time define a minimal time to let gate opened.

XINPUT section will let you use up to 4 game pad to control a button.

HID Device Button allows connecting directly to a specific HID device (to be implemented).

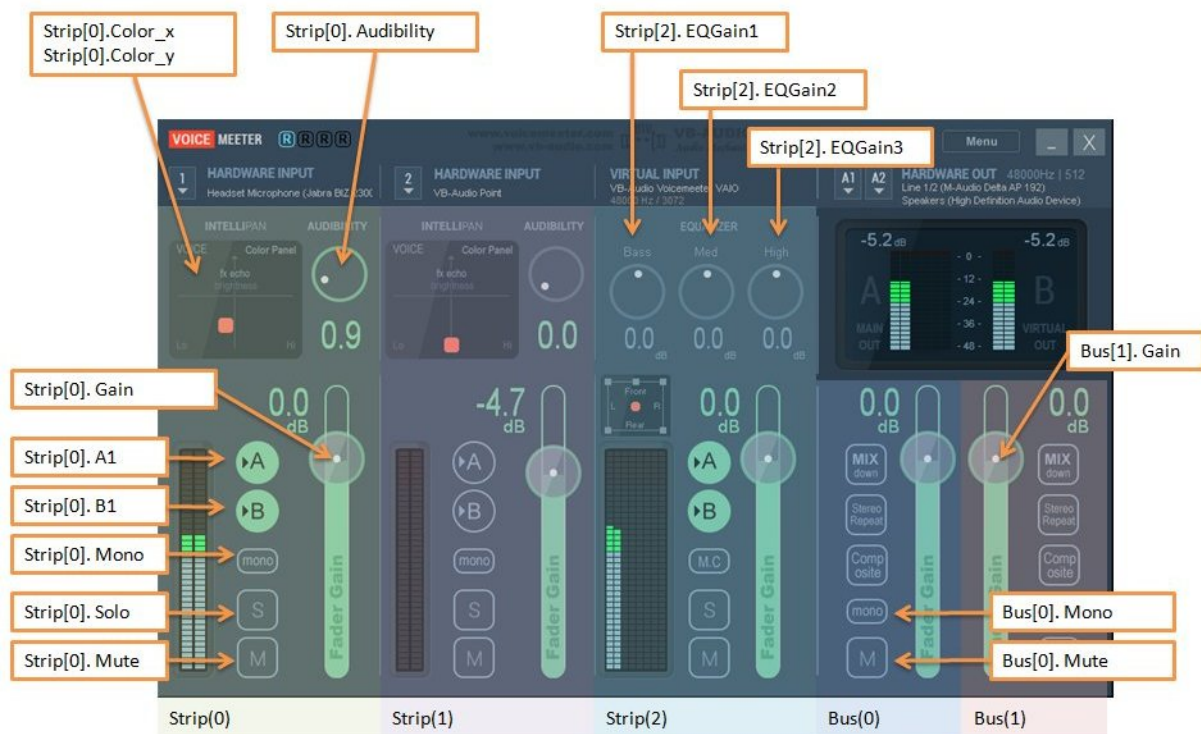
Button Color

The parameter called "Button Color" allows selecting 8 other background colors for the button.



Voicemeeter Remote Requests

Request is composed by a structured name (related to voicemeeter control or parameter) and a value or a string behind.



Request example with numbers:.

```
Strip(0).mute=1;           // MUTE ON the first strip of Voicemeeter
Strip(0).mute=0;           // UNMUTE the first strip of Voicemeeter
Strip(0).mute +=1;         // Change the current state of the Mute button

Bus(0).mono=1;             // Set the first BUS in Mono mode

Bus(0).gain=-10.0;         // set BUS Slider Gain to -10.0 dB
Strip(0).gain=+6.0;        // set Strip Slider Gain to +6.0 dB

Bus(0).gain +=3.0;         // Add 3 dB to current BUS Slider Gain
Strip(0).gain -=3;         // Remove 3 dB to current Strip Slider Gain

Command.Restart = 1;       // request to restart audio engine
```

Request example with string:.

```
Command.Load= "C:\My Documents\VMConfig1.xml"; //load Config File
```

Input Strip Parameters:

Strip index is a zero based index related to Voicemeeter version (3 strips on Voicemeeter, 5 on Voicemeeter Banana)

Strip functions/parameters

Parameter Name	Value Range	Remark	Ver.
Strip[i].Mono	0 (off) or 1 (on)	Mono Button	1
Strip[i].Mute	0 (off) or 1 (on)	Mute Button	1
Strip[i].Solo	0 (off) or 1 (on)	Solo Button	1
Strip[i].MC	0 (off) or 1 (on)	Mute Center Button	1
Strip[i].Gain	-60 to +12 db	Gain slider	1
Strip[i].Pan_x	-0.5 to +0.5		1
Strip[i].Pan_y	0 to 1.0		1
Strip[i].Color_x	-0.5 to +0.5	Physical Strip Only	1
Strip[i].Color_y	0 to 1.0	Physical Strip Only	1
Strip[i].Audibility	0 to 10	Voicemeeter 1 only	1
Strip[i].EQGain1	-12 to +12 db	Virtual Strip Only	1
Strip[i].EQGain2	-12 to +12 db	Virtual Strip Only	1
Strip[i].EQGain3	-12 to +12 db	Virtual Strip Only	1
Strip[i].Label	String	Strip Label	1
Strip[i].A1	0 (off) or 1 (on)	Out BUS Assignment	1
Strip[i].B1	0 (off) or 1 (on)	Out BUS Assignment	1
Strip[i].FadeTo	String	(dBTarget, msTime);	1
Strip[i].FadeBy	String	(dB relativechange, msTime);	1

i= strip zero based index. J= Bus zero based index.

Strip Audio Devices (physical strip only)

Parameter Name	Value Range	Remark	Ver.
Strip[i].device.wdm	Device Name	String	1
Strip[i].device.ks	Device Name	String	1
Strip[i].device.mme	Device Name	String	1
Strip[i].device.asio	Device Name	String	1

i= strip zero based index.

BUS Parameters:

Bus index is a zero based index related to Voicemeeter version (2 busses on Voicemeeter, 5 on Voicemeeter Banana)

Bus functions/parameters

Parameter Name	Value Range	Remark	Ver.
Bus[i].Mono	0 (off), 1 (mono) 2 (stereo reverse)	Mono Button	1
Bus[i].Mute	0 (off) or 1 (on)	Mute Button	1
Bus[i].Gain	-60 to +12 db	Gain slider	1
Bus[i].mode.normal	0 (off) or 1 (on)	BUS Mode	1
Bus[i].mode.Amix	0 (off) or 1 (on)	BUS Mode	1
Bus[i].mode.Repeat	0 (off) or 1 (on)	BUS Mode	1
Bus[i].mode.Composite	0 (off) or 1 (on)	BUS Mode	1
Bus[i].FadeTo	String	(dBTarget, msTime);	1
Bus[i].FadeBy	String	(dB change,msTime);	1

i= bus zero based index.

BUS Audio Devices (physical bus only)

Parameter Name	Value Range	Remark	Ver.
Bus[i].device.wdm	Device Name	Write only	1
Bus[i].device.ks	Device Name	Write only	1
Bus[i].device.mme	Device Name	Write only	1
Bus[i].device.asio	Device Name	Write only	1

i= bus zero based index.

Special functions to make timed fade in/out:

Strip().FadeTo or Bus().FadeTo function allow to set the gain slider with a progressive fade by settings a dB value and a time to reach it (time in ms between 0 and 120000). The parameter is a string since it needs 2 parameters, example:

```
Strip(0).FadeTo= (-10.0, 500); //will set the slider to -10 dB in 500ms
Strip(0).FadeTo= (-20.0, 2000); //will set the slider to -10 dB in 2 seconds
Bus(0).FadeTo= (0.0, 1500); //will set the bus slider to 0 dB in 1,5 seconds
```

System Settings Option:

Voicemeeter remote API also allows changing different configuration parameters: Patch and System Settings.

Patch Options

Parameter Name	Value Range	Remark	Ver.
patch.asio[i]	0 to ASIO input	ASIO Patch	1
Patch.composite[j]	0 to 22 (1 = first channel)	0 = default BUS	2
Patch.insert[k]	0 (off) or 1 (on)	Virtual ASIO insert	2
Patch.PostFaderComposite	0 (PRE) or 1 (POST)	COMPOSITE Mode	2
Patch.PostFxInsert	0 (PRE) or 1 (POST)	Virtual INSERT Point	2

i= input channel zero based index (for physical strips only – 2 channels per strip).

j= composite channel zero based index (0 to 7) COMPOSITE mode is made of 8 channels.

k= input channel zero based index (0 to 21).

System Settings

Parameter Name	Value Range	Remark	Ver.
Option.sr	44.1, 48, 88.2, 96, 176.4 or 192 kHz	Preferred samplerate	1
Option.ASIOsr	0: default ASIO Samplerate 1: preferred samplerate.	For ASIO driver connected on output A1	1
Option.delay[i]	0 to 500ms max	BUS output delay	1
Option.buffer.mme	128 to 2048	MME buffer size	1
Option.buffer.wdm	128 to 2048	WDM buffer size	1
Option.buffer.ks	128 to 2048	KS buffer size	1
Option.buffer.asio	128 to 2048	ASIO Buffer Size	1
Option.mode.exclusif	0 (off) or 1 (on)	WDM input exclusive	1
Option.mode.swift	0 (off) or 1 (on)	WDM swift mode	1

i= output zero based index (for physical bus only)

Special Commands:

Special Commands are not made to change parameter but to make an action. Here some command presented in the menu (write only of course).

Command Name	Value Range	Remark	Ver.
Command.Shutdown	1	Shutdown Voicemeeter	1
Command.Show	1	Show Voicemeeter	1
Command.Restart	1	Restart Audio Engine	1
Command.Eject	1	Eject Cassette	1
Command.Reset	1	Reset All configuration	1
Command.Save	String	Complete filename (xml)	1
Command.Load	String	Complete filename (xml)	1

Typical use:

```
Command.Restart = 1; // request to restart audio engine
```

```
Command.Load= "C:\My Documents\VMConfig1.xml"; // load Config File
```

NOTE: command requests are priority to other requests. It means other type of request could not be processed if in the same request packet than a command request. For example Shutdown Request, simply closes the program without processing next request. LOAD request reset all possible previous or next request present in the same packet.

EJECT cassette function (also present in the menu) is made to release audio file and let other applications access to it.

Button command for button interactions

It is also possible to change the state of one or several other buttons by using the following commands, to change the state only of a button, or emulate PUSH / RELEASE function.

```
Button(5).State = 1;      //PUSH the button ID 5
Button(5).State = 0;      //RELEASE the button ID 5

Button(5).StateOnly = 1;  //set the button ID 5 in pushed state
```

This instruction below works only in INIT Script:

```
Button.State = 0;         //change the state of the current button
```

Wait command to create sequence of requests

With the September 2019 version, the Wait command allows to introduce timing pause between requests and finally create a sequence of requests.

```
Strip(0).gain=-12.0; // will be send on button push
Wait(2000);
Strip(0).gain=0.0;   // will be sent 2 seconds later
Wait(1000);          // wait one second again
Strip(0).FadeTo= (-10.0, 1000);
Wait(1000);
Strip(0).FadeTo= (0.0, 1000);
```

Load Button map

With the Mars 2020 version, the Load command allows to use a button to load another Button Map (MacroButtons config file) previously stored with the SAVE function in the system menu of the eMacroButtons application.

```
Load("filename");
```

VBAN Options:

Voicemeeter remote API allows controlling VBAN features and all parameters presented on the VBAN dialog Box. Then it is possible to remote VBAN functions to route/send/receive audio to/from different computers.

VBAN Options

Parameter Name	Value Range	Remark	Ver.
<code>vban.Enable</code>	0 (off) or 1 (on)	VBAN functions	1
<code>vban.instream[i].on</code>	0 (off) or 1 (on)	Stream On/Off	1
<code>vban.instream[i].name</code>	String	Stream Name	1
<code>vban.instream[i].ip</code>	String	IP Address from	1
<code>vban.instream[i].port</code>	16 bit range	PORT (Ethernet)	1
<code>vban.instream[i].sr</code>	11025 to 96 kHz	Read only	1
<code>vban.instream[i].channel</code>	1 to 8	Read only	1
<code>vban.instream[i].bit</code>	VBAN data type	Read only	1
<code>vban.instream[i].quality</code>	0 to 4	0 = Optimal	1
<code>vban.instream[i].route</code>	0 to 8	Strip Selector	1
<code>vban.outstream[i].on</code>	0 (off) or 1 (on)	Stream On/Off	1
<code>vban.outstream[i].name</code>	String	Stream Name	1
<code>vban.outstream[i].ip</code>	String	IP Address To	1
<code>vban.outstream[i].port</code>	16 bit range	PORT (Ethernet)	1
<code>vban.outstream[i].sr</code>	11025 to 96 kHz		1
<code>vban.outstream[i].channel</code>	1 to 8		1
<code>vban.outstream[i].bit</code>	VBAN data type	1 = 16 bits PCM	1
<code>vban.outstream[i].quality</code>	0 to 4	0 = Optimal	1
<code>vban.outstream[i].route</code>	0 to 8	BUS selector	1

i= zero based index (0 to 7).

Like in the VBAN Dialog Box the following parameters change are producing an Audio Engine Restart:

- `vban.Enable`
- `vban.instream[i].port`
- `vban.instream[i].quality`
- `vban.outstream[i].quality`

VBAN SampleRate:

11025, 16000, 22050, 24000, 32000, 44100, 48000, 64000, 88200, 96000 Hz

VBAN Quality:

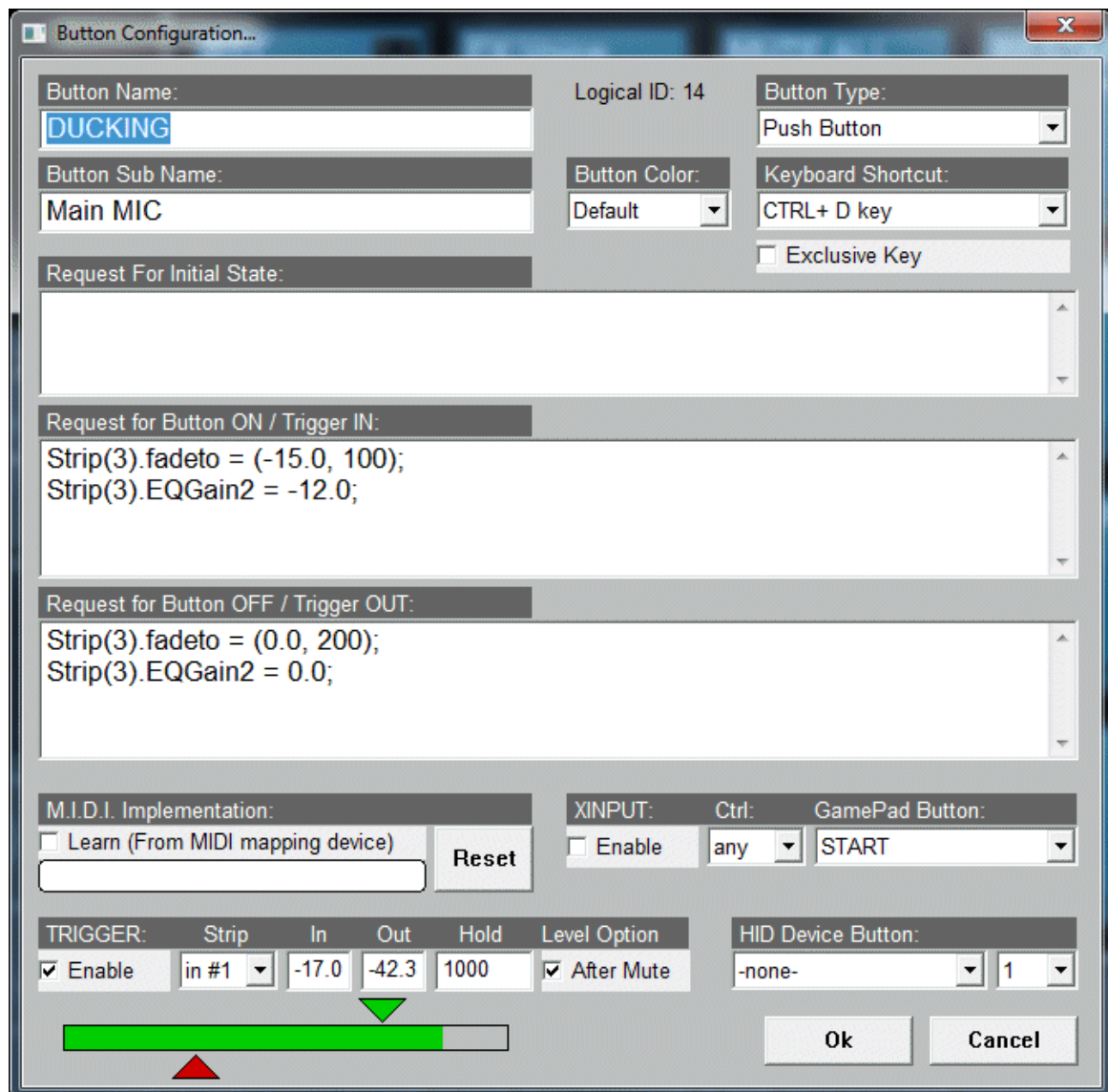
0 (Optimal), 1 (Fast), 2 (Medium), 3 (Slow), 4 (very slow). Quality parameter is conditioning the size of internal stack (also the latency) to cope to possible network instability and then increase the stream stability if required. **Optimal** quality considers the network is able to transmit packet in real time (with a good regularity), while **very slow** quality considers the network can have timing problem and unexpected waiting cycles. This parameter is more useful on receiver side while the transmitter is supposed to work always in OPTIMAL mode.

VBAN Bit Resolution / data format:

Allowed Format are 1 (16 bit PCM) or 2 (24 bits PCM).

AUTO Ducking (Trigger):

With the TRIGGER, it is possible to make auto ducking and control any parameters or function (REC button for example) according an input level.



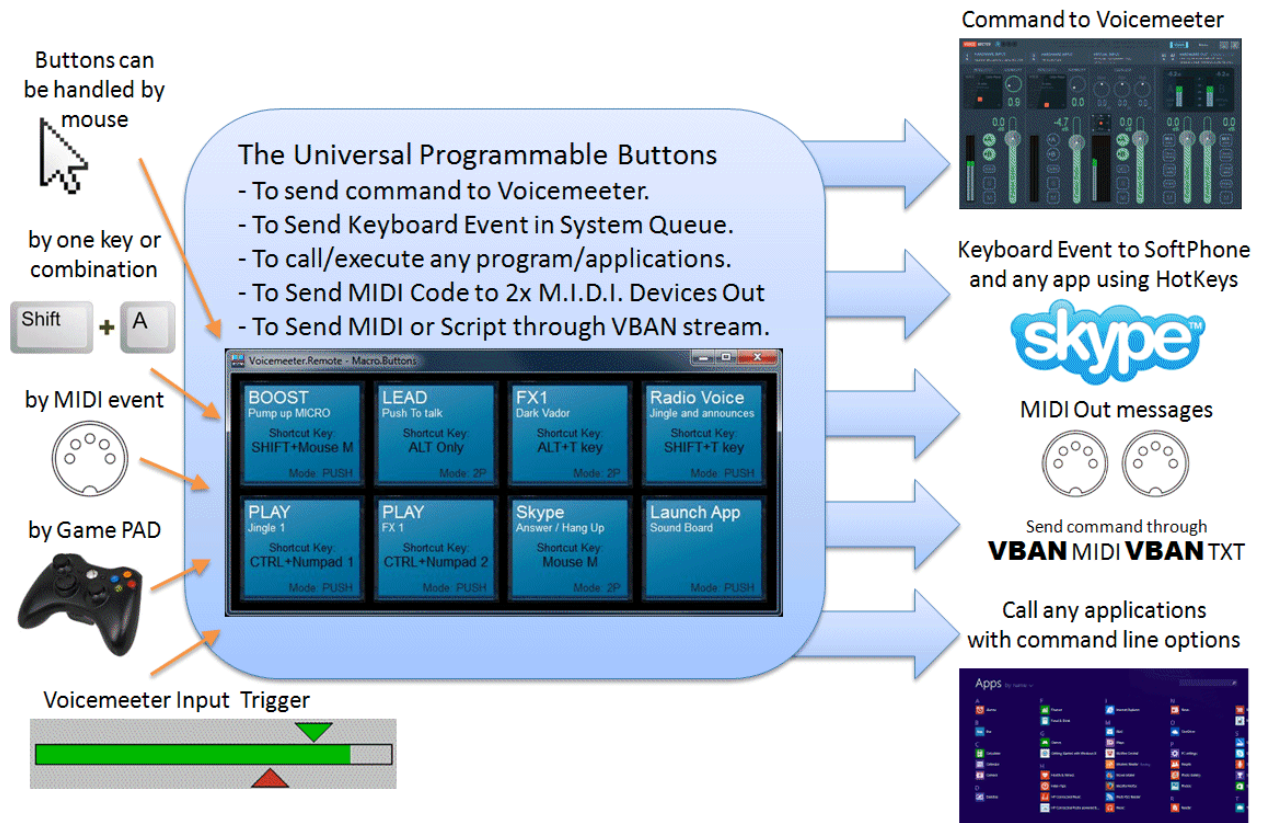
In this example, the hardware input in#1 is the Microphone and the Strip(3) is the virtual input of Voicemeeter Banana (used to get music). This virtual input will be faded to -15 dB and medium EQ by -12 dB when the user will talk in the microphone, more precisely, when the microphone level will go over -13 dB (to be adjusted according microphone level).

“Level Option” allows user to mute his microphone on Voicemeeter to disable the trigger. Otherwise trigger is working even if the microphone is muted on Voicemeeter.

Green cursor shows the “IN” threshold (level to go over to generate a trigger in). Red cursor shows the “OUT” threshold (level to go below to generate trigger out).

System Functions (to send command to Windows):

In Macro Buttons application, some instructions are made for operating system to send Keyboard Event or run “command line” like program. These special instructions, following a “function” like syntax, can be inserted in the request script, with instructions for Voicemeeter.



System Command

function Name	Value Type	Remark	Ver.
System.KeyDown(szKey)	String		1
System.KeyUp(szKey)	String		1
System.KeyPress(szKey)	String	Send Key Down + Key Up	1
System.Execute(exe, dir, arg)	Strings		1

These commands are not sent to Voicemeeter but directly to operating system.

System Execute

This function works like a “CreateProcess” or ShellExecute under windows and allow to start any application with a command line argument.

```
System.Execute(szprogram, szworkdir, szcommand);
```

Example to open a web page with the internet explorer:

```
System.Execute("C:\Program Files\Internet Explorer\iexplore.exe",  
              "-new www.voicemeeter.com");
```

Special chars like double quotes can be inserted by this sequence %' (percent + simple quote): then %' will be replaced by “. To insert a percent char, simply enter it double: then “%%” will be replaced by a single ‘%’.

ENVIRONMENT VARIABLES

It is also possible to use system environment variable by using **%envname%** syntax.

Example to run the Microsoft WRITE Editor application

```
System.Execute("%windir%\write.exe", "%TMP%", "");
```

RUNNING DOS APPLICATION

To run command line program, you need to launch cmd.exe with /K command to specify you want to execute the command after...

Example to run ipconfig in a DOS window:

```
System.Execute("%windir%\system32\cmd.exe", "%windir%\system32", "/K ipconfig");
```

/C Carries out the command specified by string and then terminates

/K Carries out the command specified by string but remains

Example to ping your internet router (usual address is 192.168.1.1):

```
System.Execute("%windir%\system32\cmd.exe", "%windir%\system32", "/K ping 192.168.1.1");
```

System KeyDown / KeyUp / KeyPress

This functions allow to send a combination of 1 to 4 keys by a simple string describing this keyboard combination, like "CTRL+SHIFT + F10" or simply "0".

```
System.KeyDown(szKey);  
System.KeyUp(szKey);
```

Example:

```
System.KeyDown("A");  
System.KeyDown("SHIFT+T");
```

```
System.KeyUp("A");  
System.KeyUp("SHIFT+T");
```

```
System.KeyDown("CTRL+NP1");  
System.KeyDown("ALT+F8");
```

```
System.KeyUp("CTRL+NP1");  
System.KeyUp("ALT+F8");
```

KeyPress function send Down and UP message in a single function.

```
System.KeyPress("CTRL+NP1");  
System.KeyPress("ALT+F8");
```

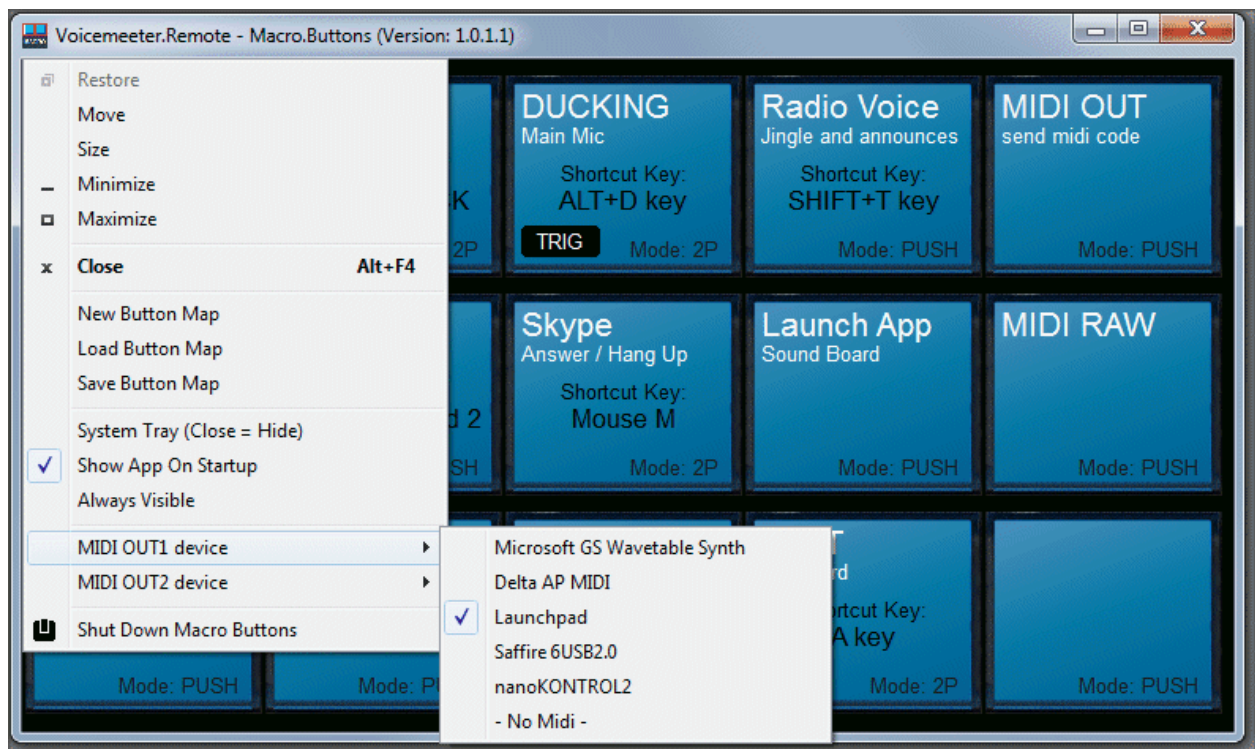
List of Key Name:

Regular Keys	NUM PAD	Special Key	FUNCTION
0 to 9	NP0 to NP9	BROWSERBACK	SHIFT
A to Z	NPMUL	BROWSERFORWARD	CTRL
BACK	NPADD	BROWSERREFRESH	ALT
TAB	NPDOT	BROWSERSTOP	
RETURN	NPSUB	BROWSERSEARCH	LWIN

ESC	NPDEC	BROWSERFAV	RWIN
SPACE	NPDIV	BROWSERHOME	LSHIFT
PAGEUP	NUMLOCK	VOLUMEMUTE	RSHIFT
PAGEDOWN	SCROLLLOCK	VOLUMEDOWN	LCTRL
END	CAPSLOCK	VOLUMEUP	RCTRL
HOME	PRINTSCREEN	MEDIANEXT	LMENU
LEFT	PAUSE	MEDIAPREV	RMENU
UP	CLEAR	MEDIASTOP	
RIGHT	SELECT	MEDIAPAUSE	F1 to F12
DOWN	PRINT	LAUNCHMAIL	F13 to F24
INSERT	PRINTSCREEN	MEDIASELECT	
DELETE	HELP	LAUNCHAPP1	
	APP	LAUNCHAPP2	
	EXECUTE	PLAY	

Send M.I.D.I. Message

In Macro Buttons 1.0.1.1 it is possible to select 2x MIDI output devices, defining out1 and out2 MIDI device. Then it is possible to send MIDI messages to these devices by a simple script.



4x request types are implemented (channel 1 to 16):

```
System.SendMidi("out1", "note-on", channel, note, velocity);
System.SendMidi("out1", "note-off", channel, note, velocity);
System.SendMidi("out1", "ctrl-change", channel, ctrl, value);
System.SendMidi("out1", "prg-change", channel, nPrg);
```

There is also a RAW DATA function to send any M.I.D.I. Message (including sys-ex).

```
System.SendMidi("out1", "data", aa, bb, cc, ee, ff, gg, ...);
```

(In this function only, values are in HEXA-Decimal: 00 to FF)

Note that values are in decimal (0 to 127) in MIDI function except for data message, value are in hexa decimal (16 based 0 to F / e.g. 90 = 9 x 16 + 0 = 144)

Button Configuration...

Button Name: MIDI OUT

Button Sub Name: send midi code

Logical ID: 4

Button Type: Push Button

Button Color: Default

Keyboard Shortcut: Numpad 2

☐ Exclusive Key

Request For Initial State:

```
System.SendMidi("out1", "note-on", 1, 0, 1);
System.SendMidi("out1", "note-on", 1, 16, 16);
```

Request for Button ON / Trigger IN:

```
System.SendMidi("out1", "note-on", 1, 0, 1);
System.SendMidi("out1", "note-on", 1, 16, 16);
System.SendMidi("out1", "data", 90, 20, 30, 90, 30, 30);
```

Request for Button OFF / Trigger OUT:

```
System.SendMidi("out1", "note-off", 1, 0, 0);
System.SendMidi("out1", "note-off", 1, 16, 0);
System.SendMidi("out1", "data", 80, 20, 0, 80, 30, 0);
```

M.I.D.I. Implementation:

☐ Learn (From MIDI mapping device) **Reset**

XINPUT: ☐ Enable Ctrl: any GamePad Button: START

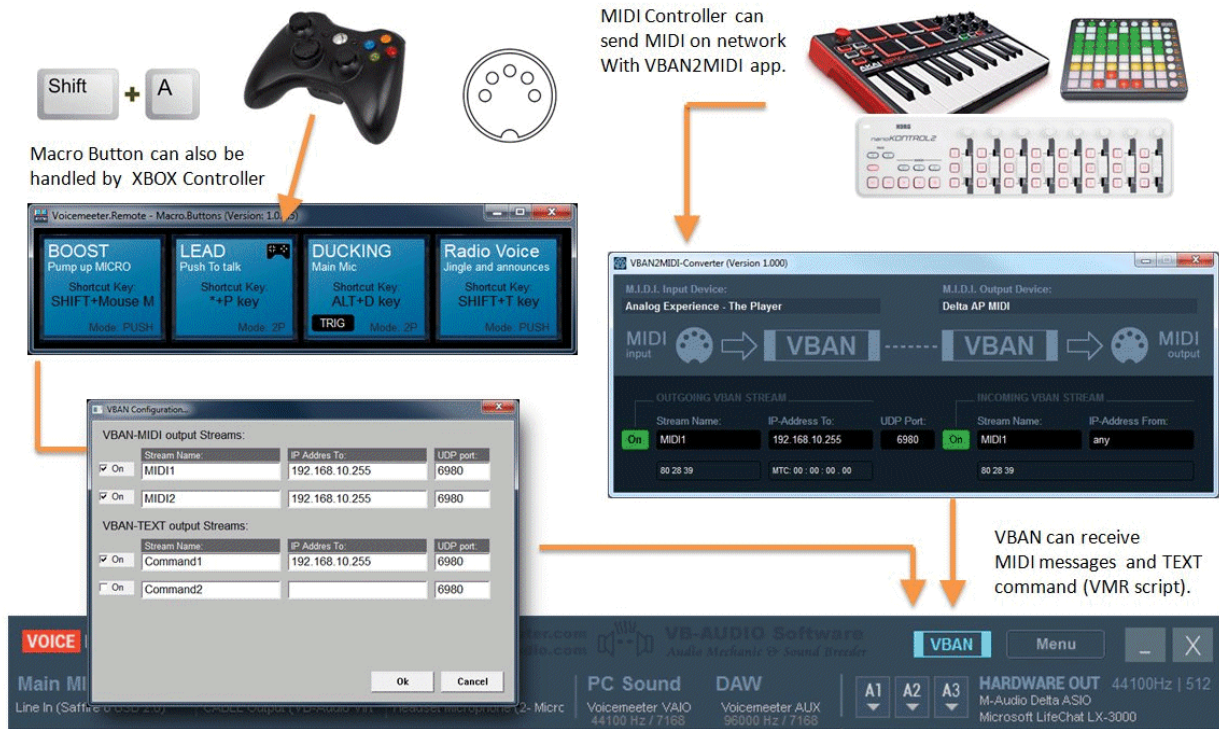
TRIGGER: ☐ Enable Strip: in #1 In: 0.0 Out: 0.0 Hold: 100 Level Option: ☐ After Mute

HID Device Button: -none- 1

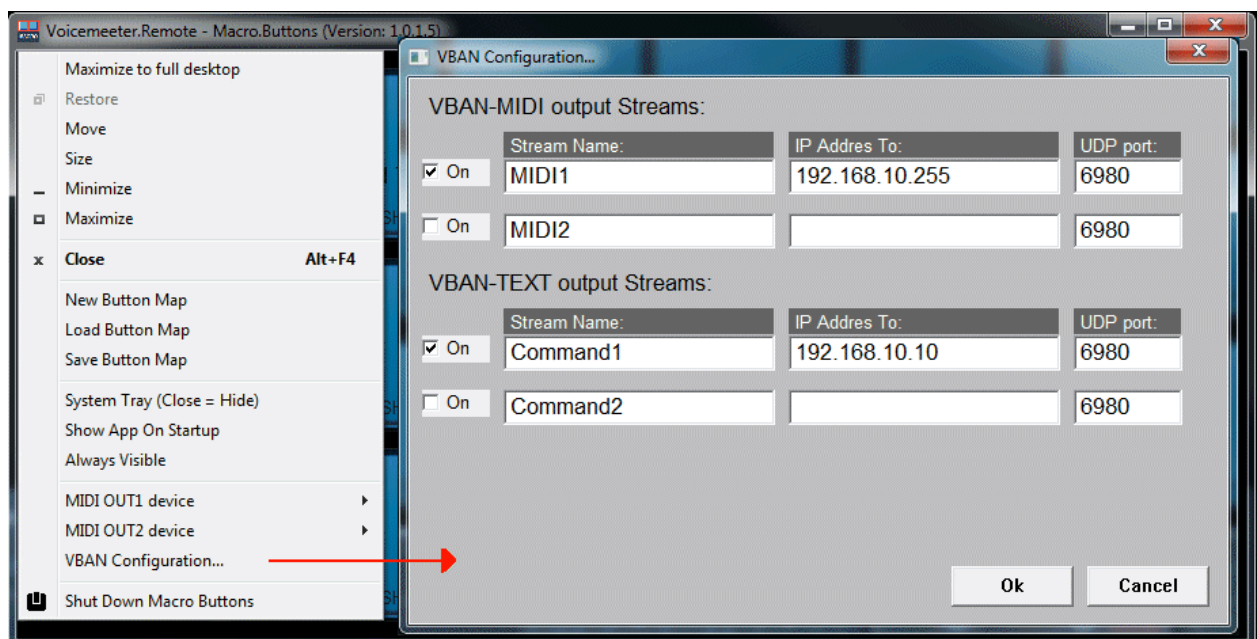
Ok Cancel

Send VBAN-MIDI or VBAN-TEXT (Voicemeeter script)

With Voicemeeter version 1.0.3.5 / 2.0.3.5 it is possible to send MIDI message through VBAN-MIDI and Voicemeeter Script through VBAN-TEXT protocol. And the MacroButton application can also learn MIDI code coming from incoming VBAN-MIDI stream. Note that MIDI2VBAN application is also installed with Voicemeeter to convert physical MIDI I/O into VBAN-MIDI Stream.



To send message through VBAN protocol, you have first to activate and configure your VBAN output streams. This is located in MacroButton system menu: VBAN Configuration Dialog Box will allow to setup 2x VBAN MIDI streams and 2x VBAN TXT Streams:



Send MIDI command through VBAN Stream

Once you have configured output VBAN MIDI Stream, you can use the same instruction to send MIDI message to VBAN1 or VBAN2 stream:

4x request types are implemented (channel 1 to 16):

```
System.SendMidi("vban1", "note-on", channel, note, velocity);
System.SendMidi("vban1", "note-off", channel, note, velocity);
System.SendMidi("vban1", "ctrl-change", channel, ctrl, value);
System.SendMidi("vban1", "prg-change", channel, nPrg);
```

There is also a RAW DATA function to send any M.I.D.I. Message (including sys-ex).

```
System.SendMidi("vban1", "data", aa, bb , cc, ee, ff, gg, ...);
```

Send Voicemeeter TEXT Request through VBAN Stream

Once you have configured output VBAN TEXT Stream, you can send a command script by using a section bloc starting by BEGIN_SECTION and ending by END_SECTION instruction:

```
BEGIN_SECTION("vban1")
Strip(0).mute=1; Strip(1).mute=1;
Bus(0).gain= 0.0;
END_SECTION
```

All Voicemeeter script written inside the section will be sent to VBAN1 TEXT Stream defined in the VBAN configuration Dialog Box. While the section below will send the script through VBAN2 TEXT stream:

```
BEGIN_SECTION("vban2")
Strip(0).mute=1; Strip(1).mute=1;
Bus(0).gain= 0.0;
END_SECTION
```

Regular function style syntax is also possible (to fit the syntax used in VBAN-Button app for Android):

```
SendText("vban2",Strip(0).mute=1; Strip(1).mute=1;Bus(0).gain= 0.0);
```

Or written on several lines as well: last parenthesis will close the entire request.

```
SendText("vban2",
Strip(0).mute=1;
Strip(1).mute=1;
Bus(0).gain= 0.0);
```

Manage your lights Network in DMX 512

With Macro Buttons 1.0.2.7 it is possible to send DMX 512 requests through a DMX serial interface (COM) selected in DMX configuration dialog box in MacroButtons system menu by a simple set of 2 functions:

The simple function to set a DMX value according device address and channel:

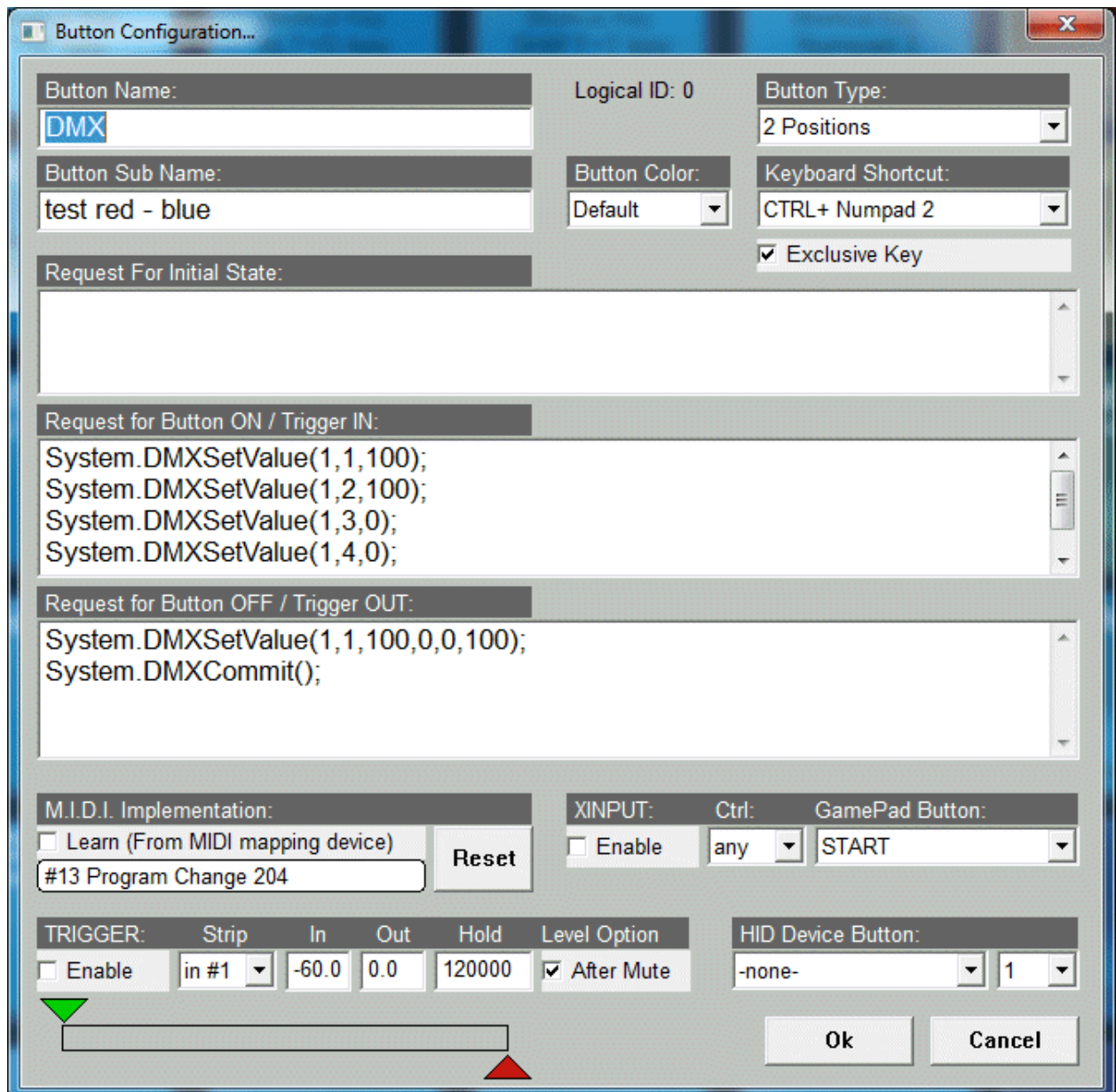
```
System.DMXSetValue(addr, channel, value);
```

or can be used with several values which will automatically fill the next channels:

```
System.DMXSetValue(addr, channel, value1, value2, value3, value4...);
```

To send the newly modified DMX frame:

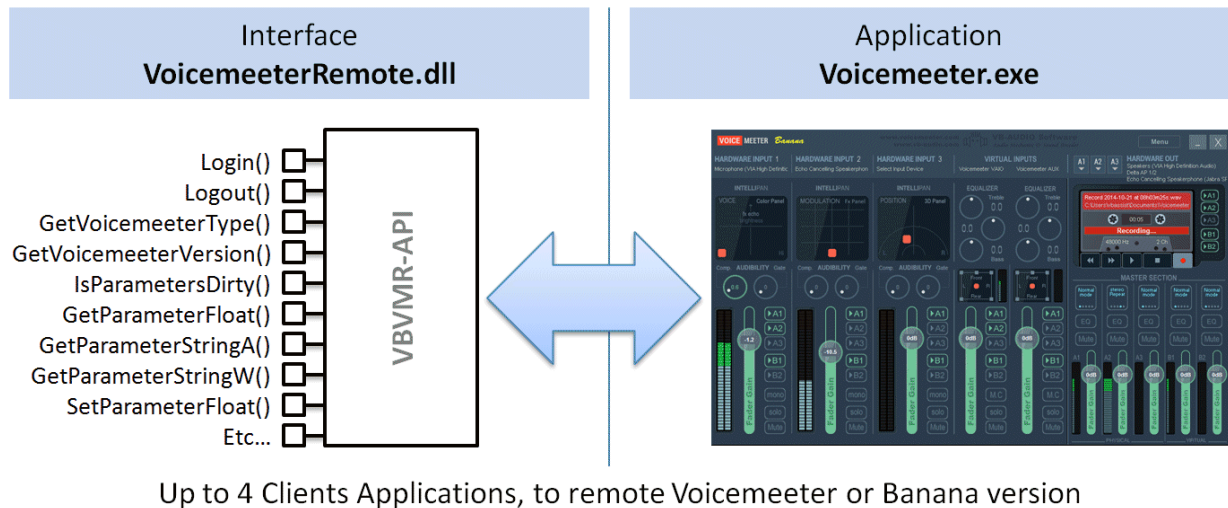
```
System.DMXCommit();
```



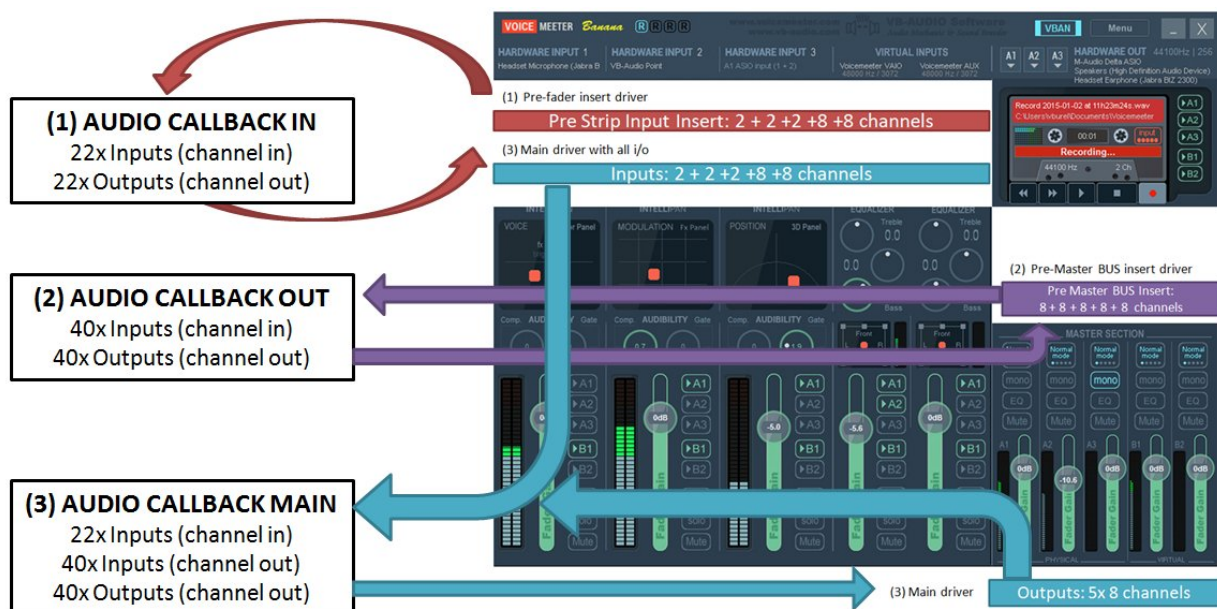
Example of tested DMX serial interface: Enttec Open DMX USB Interface

Voicemeeter Remote API (for developer only)

The requests described above are based on Voicemeeter remote API (delivered and installed with Voicemeeter as VoicemeeterRemote.dll). This API can be used by any third part application programmed in any language to control Voicemeeter and take advantage of all its features.



Since Voicemeeter 1.0.5.0 / 2.0.3.0, Voicemeeter Remote API provides an AUDIO API to process audio inside Voicemeeter in 3 Different point. Download SDK and get more information on our forum: <https://forum.vb-audio.com/viewtopic.php?f=8&t=346>



EXTRA OPTIONS

Voicemeeter Latency

M.I.D.I. Mapping

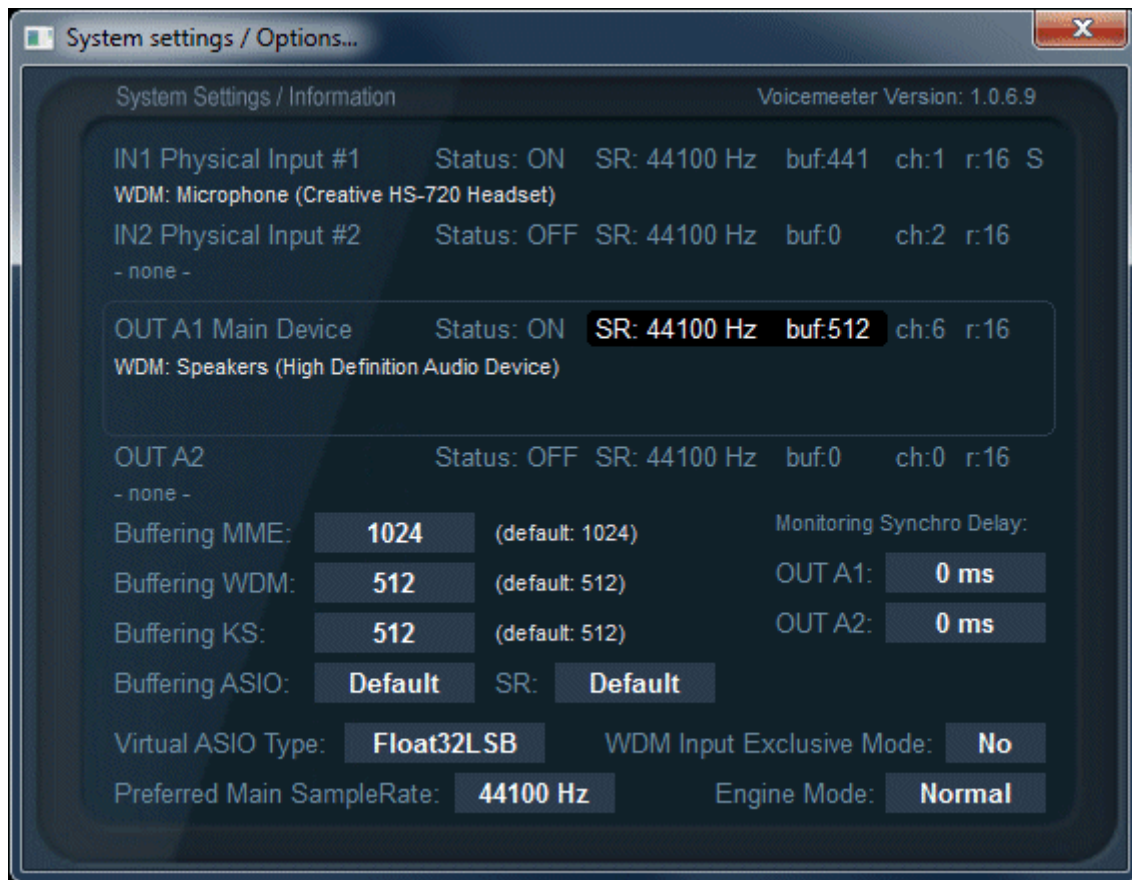
Specifications

Known issues

Command line Installation

System Settings / Options

Click on Master Meter LCD Section to open the Options Dialog Box.



This modal dialog box will show you the status of the different audio devices currently used by Voicemeeter.

- **SR**: is giving the current samplerate of the device (which can be different for each device, Voicemeeter is based on multi points and multi formats audio engine able to manage most of audio system configurations).
- **buf** gives the current buffer size used by the audio device (Voicemeeter Audio engine support different buffering on all audio points).
- **Ch**: gives the number of channel managed by the devices (Voicemeeter can manage 1 to 2 channels on inputs and up to 8 channels on outputs).
- **r**: gives the bit resolution of the device (Voicemeeter uses 16bits resolution per default).
- **S**: means "share mode" (applicable for WDM device only. KS is usually in exclusive mode and MME in share mode. For WDM it depends on devices and audio engine mode).

Note: Output A1 device configuration gives the main audio configuration of Voicemeeter and the configuration of Voicemeeter virtual ASIO driver (Sample rate and buffer size). It also means that all audio processing done in Voicemeeter will be based on this sample rate (if some audio points work in another sample rate, conversion will be done automatically to this main sample rate.).

In this dialog box, you can change the latency used by MME, WDM and/or KS audio drivers. Basically MME driver support buffering between 512 and 2048 samples , while WDM and KS

can go down to 256 samples (that makes audio processing very closed to the real time – practically usable to sing on a song in real time – karaoke – or to play digital piano on music in real time).

Some PC configurations (or some audio devices) will work very well with minimal buffering values while other will not (the sound stream could be not stable and generates too much sound's cut). That's why per default MME device uses 1024 samples buffering while WDM device is using 512 samples (these default settings should work for 100% PC configuration cases).

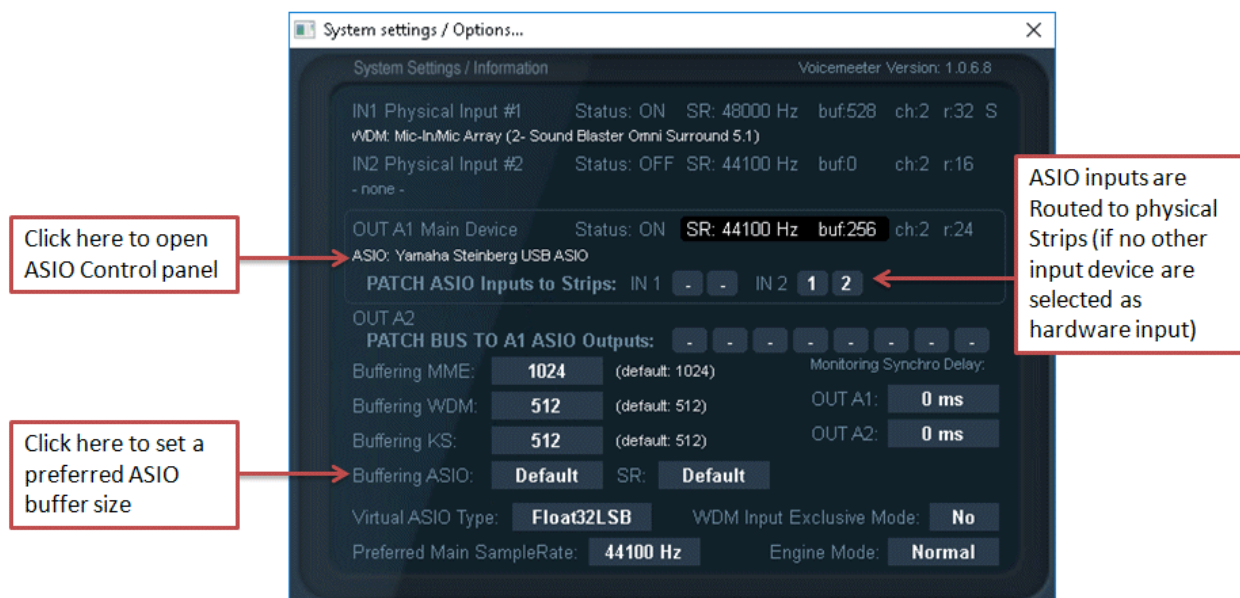
Under Windows XP, WDM driver are not available. (That we call WDM driver is audio device driver that can be handled by Microsoft WASAPI, for better audio performances and quality). But some audio device can support KS interface and then provide low latency support.

Preferred Main Sample Rate can be used as a wish to make Voicemeeter run output device A1 in the wanted sample rate (44.1, 48, 88.2, 96, 176.4 kHz or 192 kHz). But it also depends on the audio device current configuration (that could stay in its own sample rate anyway).

Engine mode provides a SWIFT mode to possibly improve real time (experimental option). WDM Input devices are driven in SHARED mode per default to avoid possible WASAPI Exclusive mode bug (see known issues).

ASIO Driver support:

Voicemeeter can use an ASIO driver as Output A1 (the main output). Then Voicemeeter will run in audio pro conditions (like any DAW using ASIO device).



If ASIO device is selected as output A1, Voicemeeter will send audio to the 8 first output channels. Other BUS could be routed to other ASIO output channels. It is also possible to get the first input channels of the ASIO device routed to Hardware Inputs of Voicemeeter. This is done by the ASIO PATCH allowing to select which ASIO channel will be used for IN#1 left & right, IN#2 left and right...

ASIO Buffering and ASIO samplerate are pending on two parameters:

- Buffering ASIO can use the default buffer size (given by ASIO driver) or be forced to a given buffer size (but the ASIO driver can refuse or not work correct for some values).
- SR parameter allows to use the default sample rate given by the ASIO driver or to force the sample rate defined by the preferred sample rate.

It is also possible to open the ASIO Control panel of the main output device by clicking on its name.

Getting Optimal Latency:

The choice of the Output A1 is critical to get the best global latency, since it gives the master sample rate and main buffer size. So we recommend to select your best audio device as output A1 (preferred ASIO if possible, WDM or KS otherwise) – with ASIO device the buffer size is given by the ASIO driver itself (that can usually be set by the ASIO Specific Control Panel of your audio board). But it is possible to specify a preferred ASIO buffer size (used if the driver accepts it).

To improve latency, you can also decrease the latency buffer size (for example down to 256) especially with WDM or KS devices. If using WDM devices, you can again reduce global latency (the time between microphone and speaker) by using SWIFT mode (but not recommended because might be unstable).

Activating WDM Input Exclusive mode (and Swift mode) will force audio engine to use smallest buffer with WDM input devices. But this exclusive mode could not be stable and produce various potential problems in the time (from audio crash to system freeze/crash – pending on audio driver and O/S – see known issues section at the end of this document).

Virtual I/O latency can also be optimized by decreasing the Internal Latency of the Voicemeeter VAIO driver by using specific VBCABLE Control Panel program. However decreasing internal latency can produce discontinued or non-working stream in some cases, pending on different buffering constraints (see VB-CABLE technical documentation).

Virtual ASIO driver are adding a single buffer to the global latency (the size of this buffer is given by the buffering of the output A1)

LATENCY WARNING:

CHANGING DEFAULT LATENCY, BY REDUCING BUFFER SIZE CAN DECAY THE AUDIO STREAM, BRING UNSTABILITY, FREQUENT AUDIO CUT, STATIC, SYNCHRO LOST (ROBOTIC VOICE).

If you get such problem, come back to default buffer size.

M.I.D.I. Mapping:

Voicemeeter allows connecting a MIDI remote surface to control gain, mute, solo buttons of every strips and busses.



You just need to select the right MIDI device and enter in Learn process to configure the M.I.D.I. Mapping. This is expected to work with any M.I.D.I. Remote. You can even give a title to your mapping and store it on disk or recall it from XML files.

Click on Learn (use TAB or up/down arrow key to learn next/previous control) and move your M.I.D.I. Control.

Click on M.I.D.I. Code area (black) to reset the M.I.D.I. Implementation (as it was before learning process).

PTT option means “Push To Talk” and un-mute the related strip when pushing the button, mute it when release it.

RESET MAP: reset entire M.I.D.I. mapping

LOAD / SAVE allow to recall and save M.I.D.I. Map from or to file (xml file).

M.I.D.I. Map name is a name defined by user to identify precisely the remote surface (stored in M.I.D.I. Map XML file).

Specifications:

Device Type:	PC-Core Virtual Audio Mixing Console
Compatibility:	Windows XP, VISTA, WIN7, WIN8, WIN10 (32 / 64 bits)
PC Configuration:	Min: Celeron / Duo Core 1.8 GHz - 512 MB RAM - Disk < 100 MB
Number of Audio Device I/O:	3 Inputs (2 physicals / 1 Virtual). 3 Outputs (2 physicals / 1 Virtual)
BUS / Layer:	2x BUS (A and B) / Single Layer
Audio Engine Capabilities:	32, 44.1, 48, 88.2, 96, 176.4 or 192 kHz DSP Processing (defined by Output A1 configuration)
Output A1 (Main):	WDM, KS, MME, ASIO (32 kHz to 192 kHz) - 1 to 8 channels
Output A2:	WDM, KS, MME (8 kHz to 192 kHz) - 1 to 8 channels
2x Physical Inputs:	WDM, KS, MME (8 kHz to 192 kHz) - mono or stereo.
1x Virtual I/O:	WDM, KS, MME, DirectX, WaveRT (8 kHz to 192 kHz) 1 to 8 channels 8 channels on virtual input, 2 on virtual output
1x Virtual ASIO I/O:	ASIO (32 kHz to 192 kHz) 8 Channels (in and out) / 4x Client Applications. Virtual ASIO configuration is given by Main Output A1 (SR and Buffering)
M.I.D.I. Implementation (remoting):	Gain faders, Mute, Solo, M.C. Audiobility, 3 Bands EQ (Configuration by Learn process).
Strip Processing:	- "Color Panel" Control (Equalization). - 3D Panoramic Control (source positioning by binaural effect). - Audibility Knob (Compressor / Gate effect). - 3 Bands Graphic Equalizer (on Virtual Input). - Mute / Solo.
BUS Processing:	- Integrated 0 dBfs Limiter and Peak Remover. - Mix Down to convert 5.1 or 7.1 to Stereo. - Stereo Repeat (Stereo signal copied on ch 3,4 / 5,6 / 7,8) - Mute / Mono
Others:	- Physical Output Synchronization Delay in system settings dialog box.

Voicemeeter I/O Diagram:

Like a regular mixing console, Voicemeeter can be represented with a usual I/O Diagram even if connections are not made by real wire, principles stay the same.

Note that Virtual I/O provide Windows Interface (MME, KS, WASAPI, DirectX...) and ASIO Interface (defined by Steinberg gmbh). It allows connecting regular windows audio application as well as audio pro DAW using ASIO devices only.



KNOWN ISSUES / RECOMMENDATIONS

Voicemeeter has been validated on different levels and should work 100% in most configurations. However we have notified some few issues, mainly related to general system configuration and Windows Audio Scheme understanding.

No sound anymore on my computer!

Voicemeeter is like a mixing console, it must be connected correctly to work correctly. If you set Voicemeeter audio device as default device, Voicemeeter must be running (like switching ON your mixing console) otherwise there might be no sound on your computer. When Voicemeeter is running, it can get exclusive access to selected output device (A1, A2, A3) and prevent other applications to use these playback devices. As Virtual Audio Mixer, all audio sources should be connected to Voicemeeter to avoid possible conflict. When you connect your mixing console to your amplifier, you connect nothing else to this amplifier and all sources are connected to your mixing console. It's the same with Voicemeeter.

Voicemeeter does nothing if you don't use it

Voicemeeter application and Voicemeeter virtual audio device drivers do nothing if you don't use them and they cannot disturb or corrupt your system. If there is no sound anymore, it's because a bad audio configuration. Then you just have to check your default audio device in Windows Control Panel / Sound Dialog Box (what is your default playback or communication device? What is your default recording device ?...), and check the audio configuration of some possible application that you previously configured to work with Voicemeeter (typically Skype).

Audio Stream becomes bad (sound is choppy, discontinued):

If the audio stream is not stable in the time, with too much cut in the sound, it means your configuration does not support the current buffering latency on one of your audio device (usually the one used for output A1). In this case you can increase the buffering in Voicemeeter System Settings Dialog Box (768 or 1024 samples for WDM). If it's not enough, select MME audio device instead of WDM (especially for Output A1) because if Output A1 is not stable, it can disturb all other audio stream: Output A1 Audio device is master while all other audio points are slaves.

Output A1 and A2 are not exactly synchronized.

On Voicemeeter every i/o are independent and we can hear more or less delay between them, especially when using 2 audio outputs on BUS A: if A1 and A2 are routed to 2 audio devices, the sound might be not exactly synchronized (one speaker output can be late and produces a small echo with other speaker output). This is normal (according technical constraint) but can be corrected by compensate one audio output with a delay line (see System Settings Dialog Box).

Voicemeeter Audio Stream can be stopped (no more sound, no meter display).

It can happen when changing audio system configuration (in Windows Control Panel / Sound properties) or when plug or unplug USB device or when launching other program that could use same audio device ... In this case, simply restart audio engine or re-select an audio device out A1 or input 1 or any other in Voicemeeter. It will restart audio stream automatically (otherwise it could mean that audio device are used by other process and cannot be used anymore by Voicemeeter).

Audio Stream is stopped or corrupted after working well several hours.

If "WDM Input Exclusive Mode" is set when using WDM device as input, some PC/Audio driver configuration can be not stable and produce this unexpected behavior (including pops and cracks in the sound) after hours of perfect streaming. In this case the workaround is to disable Input Exclusive Mode in Voicemeeter System Settings (this is done by default) or use MME or KS audio device on Inputs (instead of WDM device driver).

Command Line Installation:

It is possible to make an automatic installation (or de-installation) by using command line options. However it will need to reboot after installation (and after de-installation). So it's recommended to first call the setup program to uninstall possible previous version, then reboot, and then install new version (and reboot again).

-h : to hide GUI
-i : to install
-u : to uninstall

Voicemeeter Command Line:

Voicemeeter.exe and VoicemeeterPro.exe support some command to restart audio engine or to run Voicemeeter with a pre-defined configuration file. It allows for example to add different shortcut on the desktop to run Voicemeeter for different jobs.

To Restart Audio Engine by a shortcut use the following command line.

```
Voicemeeter.exe -R (or "-r")
```

It can be useful to restart audio engine from a shortcut, for example to re-activate a known USB headset already connected to one Voicemeeter hardware input, but not physically plugged to the computer. When re-plugged to the computer, Voicemeeter Audio Engine has to be restarted to make it work again.

To run Voicemeeter with a specific configuration file, use these following command lines (Voicemeeter must have been shut down before – if Voicemeeter is already running, it will do nothing):

```
-l"filename.xml" or -L"filename.xml"
```

If the xml is in the same directories of voicemeeter.exe

```
-l"filename.xml" or -L"filename.xml"
```

If the xml is in the same sub directories of voicemeeter.exe

```
-l"subfolder\filename.xml" or -L"subfolder\filename.xml"
```

Or you can use absolute path

```
-l"c:\folder\filename.xml" or -L"c:\folder\filename.xml"
```

Registry Parameters:

Some parameters in the registry (regedit) can be interesting for some expert users in very specific use case or rare technical problems. Do not modify these values without having getting advice on our support resource (forum, mail).

We are talking here about some values present in the following registry directory

`HKEY_CURRENT_USER\VB-Audio\VoiceMeeter`

`DelayedStartS`

This value gives the countdown in second before starting audio engine on Voicemeeter Startup (5 second per default). This is represented by a growing bar with a countdown displayed on the A1, A2... device selection area when launching Voicemeeter. This is done to let the time to all audio services and drivers to be started before Voicemeeter start its audio engine. If a driver is not ready when Voicemeeter Audio engine starts, this device won't work until restarting Voicemeeter Audio Engine (Menu -> Restart Audio Engine).

`preferredResolution`

This value gives the preferred bit resolution used by Voicemeeter Banana & Potato (24 bits per default). Voicemeeter Standard is always using 16bits per default with Windows Audio Interface. In any cases, the final bit resolution is the result of a negotiation with the audio interface (that can force to use other resolution than the preferred one).

WARNING: Other Registry Value, must not be modified directly.