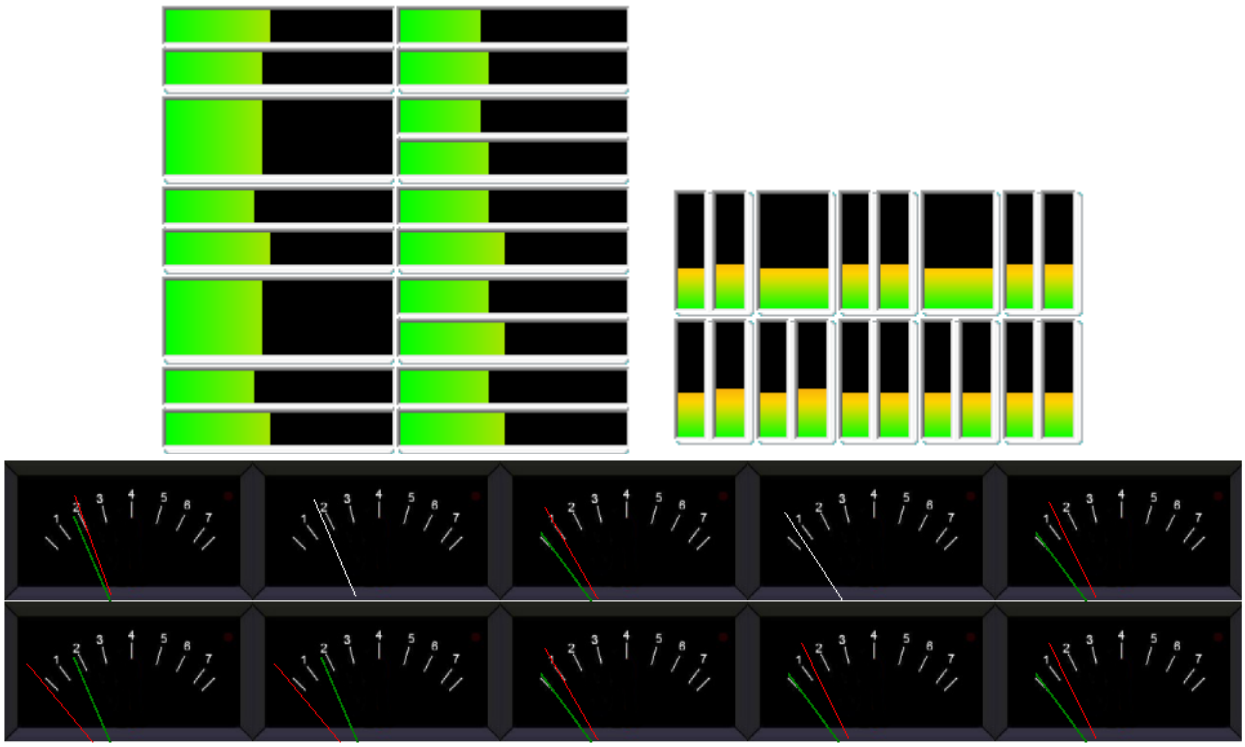
Vmeters (v2.02)



Overview

VMeters is an audio level meter application, which can read the levels of multiple audio devices, and broadcast this level info to other copies of the application on other machines on the network.

The meters can be several styles (horizontal/vertical bar meters or a PPM style meter is coming soon), can be stereo/mono and can be resized and repositioned on-screen to suit your needs. You can specify how many meters are shown, how many per row, whether to arrange them vertically or horizontally, and whether they stay on top of other apps.

VMeters can alert you if they become disconnected from their source, or if silence is detected. It can send emails, http posts (to its sister product VClock for example), and log to a file.

System Requirements

VMeters has been tested on Windows 7 (32 and 64 bit), Windows 8 (64 bit) and Windows 10 (64 bit) but should work on any modern Windows Operating System.

It requires Microsoft DotNet v4.5.2 and Microsoft DirectX (including DirectSound – which isn't always installed even if DirectX is).

Note that VMeters V2.0 onwards will NOT support XP, as XP does not support .Net 4.5.2.

Installation

Simply run the installer and answer the few questions that it asks.

Licensing

VMeters requires a license to be a “server” (ie to read values from audio devices), but as many “client” copies (ie connecting to a “server” to acquire its level information) as you like can be run without a license.

Licensing is via a VMETERS.LIC file that should be saved in the \ProgramData\Voceware\VMeters folder (or subfolder if an “instance” name is used).

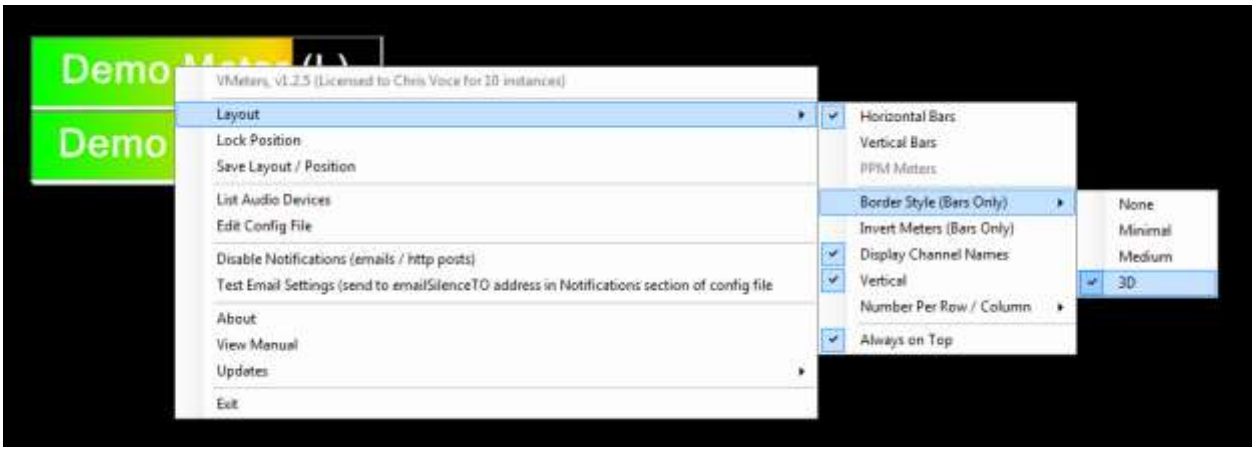
An unlicensed version acting as a server will give a warning then run for 15 minutes before it stops reading the audio levels. Client meters will continue to work.

Running multiple instances of VMeters on a single PC

If you pass a single word on the commandline (or a name with spaces if you include them in quotation marks), then VMeters will create a subfolder of \ProgramData\Voceware\VMeters when it first runs, and will store the VMeters.ini file in that folder. It will also look there for the licence file.

Control of the App

Once running, you can right-click the application to get to a menu:

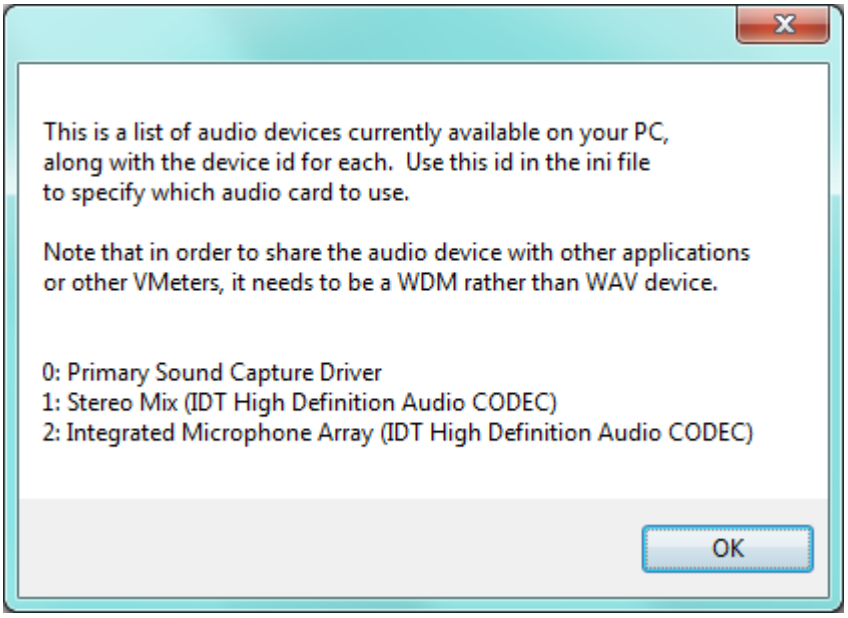


The **Layout** submenu allows you to select the style of the meters and how many meters are shown per row (or column if vertically). The **Border Style** is the border around each meter, and **Invert Meter** will make the meters display right-to-left (or top-to-bottom). **Display Channel Names** will display the channel name as text on top of the level meter. The **Vertical** checkbox selects whether to arrange the meters vertically or horizontally. **Number Per Row / Column** allows you to specify how many meters to show before starting a new row / column. **Always on Top** makes the app stay on top of other applications.

By default, the position of the app is locked on the screen, so it cannot be moved accidentally. To unlock, simply uncheck the **Lock Position** option. You can then drag the application around by clicking and dragging with the mouse. Hold down the CTRL key whilst dragging to resize the app.

Once you are happy with the new position and layout settings, simply right-click and choose the **Save Layout / Position** option.

To view a list of all audio devices available to VMeters on the machine, choose **List Audio Devices**. This will give a list, along with the channel id to use in the Source setting for each meter in the config file:



You can directly edit the config file (VMeters.ini) by choosing **Edit Config File**.

Disable Notifications, when ticked, will stop VMeters from sending email or http post notifications when silence is detected or connections are lost. It is designed for maintenance work, etc. It is the only setting that is **not** remembered when the application is restarted.

Test Email Settings will send a test email to the SilenceTo address(es) specified in the Notifications section of the config file. Note that it won't send it to addresses set in each meter section. If it fails, it will tell you the error code returned to help diagnose the issue.

You can also view an **About** box with more information about the application, or view this **manual** from the menu.

Updates: Manages the auto-update feature:

> **Check Now**: Checks www.voceware.co.uk immediately and informs you if there is a newer version, asking you if you wish to download and install it
>
> **Manual / Alert / Auto**: Set the mode of the automatic checking. VMeters will check www.voceware.co.uk once every 24 hours for a new version, unless set to manual. **Alert** will simply inform you and you will then have to "check now" to download it. **Auto** will download and apply the update automatically, restarting the application.

Finally, you can exit the application.

Other Configuration

VMeters is controlled by a simple text file, vMeters.ini, in \Programdata\Voceware\VMeters folder (or subfolder if an "instance" is specified). This has a **[settings]** section.

```
[Settings]
Position=0,0,400,300
Style=HORIZONTALBARS
Vertical=True
Inverted=False
BorderStyle=3D
NoPerRow=10
LevelFallRate=1
AlwaysOnTop=True
ShowMeterNames=True
Debug=False
BackupPath=
KeepBackups=10
```

Settings control the way VMeters looks and are mainly set by the right click menu. There are a few exceptions:

LevelFallRate is the rate at which the meter levels fall when audio is removed. A value of 1 (the default) sets the fall rate similar to that of a PPM Meter (24db in 2.7 secs, so 0.857db every 100 mS). A value of 2 will double this rate, 3 will triple, etc.

Debug, if enabled, writes a debug file (to the ProgramData\Voceware\VMeters folder (or subfolder if an "instance" is specified)), containing various information.

BackupPath is an optional path to a folder where VMeters will write a zip file containing the current configs. The file is written each time that VMeters is started and each time the config is saved. VMeters will keep the last *n* backups, *n* being set by the **KeepBackups** setting. If BackupPath is blank, the feature is disabled.

It also has information about each Meter in a **[MeterX]** section (where X increments for each meter):

```
[Meter1]
Name=Demo Meter
Source=DEMO,1
Broadcast=9001
AudioMode=STEREO
AudioOffset=0
SilenceParameters=-75,10
EmailSilenceStartSubject=VMeters - Silence detected on %NAME%
EmailSilenceEndSubject=VMeters - Audio returned on %NAME%
EmailSilenceStartExtraDetail=
EmailSilenceEndExtraDetail=
EmailLostConnectionStartSubject=VMeters - Lost Connection detected on %NAME%
EmailLostConnectionEndSubject=VMeters - Connection re-established on %NAME%
EmailLostConnectionStartExtraDetail=
EmailLostConnectionEndExtraDetail=
EmailSilenceTo=
EmailSilenceCc=
EmailSilenceBcc=
EmailLostConnectionTo=
EmailLostConnectionCc=
EmailLostConnectionBcc=
PostLocation=
PostSilenceStart=
PostSilenceEnd=
PostConnectionLost=
PostConnectionRestored=
FileLocation=
FileFormat=
Hidden=False
```

Name is a user-friendly name. If **ShowMeterNames** is set to true, this name is displayed on the meter itself. It is also used in any silence emails sent.

Source can be

- **AUDIOCARD,*n*** (ie read from an audio card, device number *n*)
    - Right-click the meters and choose "List Audio Devices" to see a list of device numbers along with the name of the device.

- **CLIENT,*ipaddress,port*** (connect to another VMeter to read level info)

- **LIVEWIRE,*ip,meter*** (connect to a Livewire (Axia) Audio Node or Console and display levels for any of its inputs or outputs). The *meter* value can be:
    - "MTR ICH *n"* (Input number *n*)
    - "MTR OCH *n*" (Output number *n*)
        - If you are not sure which input/output you are interested in, you can telnet to the Node (or Console) on port 93, then type SRC<return> (for inputs) or DST<return> (for outputs). This will give you a list of the channel numbers along with a description of what they do.

- **WHEATSTONE,ip,meter** (connect to a Wheatstone Blade and display levels for any of its inputs or outputs). The *meter* value can be:
    - "INPUT *n*" (Input number *n* (1-8))
    - "OUTPUT *n*" (Output number *n* (1-8))
    - "UTILITY *n*" (Utility Mixer *n* (1-4)… where
        - 1 = Mixer 1, Output A
        - 2 = Mixer 1, Output B
        - 3 = Mixer 2, Output A
        - 4 = Mixer 2, Output B
        - You are specifying a LEFT and RIGHT pair of meters.

- **PRONTONET,*ipaddress*,INPUT|OUTPUT** (connect to a ProntoNet codec (using the API on port 50031) and read level info for the input or output level). Note that the ProntoNet only allows a single API connection, so you can only display either the input or the output level currently. And only if the API is not already in use.

- **TIELINE,*ipaddress,port,user,password,meters*** (connect to a TieLine codec (using a web connection) and receive level information via a web stream.
    - Meters are in format c=dec0&c=dec1 (where you need to specify 2 values, the left and right meter you are interested in. Valid values are
        - c=dec0&c=dec1 – Input 1
        - c=dec2&c=dec3
        - c=dec4&c=dec5
        - c=enc0&c=enc1 … and so on up to 8

- **DEMO,*n*** (show random level info, for demo purposes only. *n* is an integer value used as the seed for the random number generator).

Broadcast is the IP Port to broadcast the level information on, for other VMeters to connect to. Leave blank to disable this feature.

AudioMode is optional (default is stereo). It can be "STEREO" (show 2 meters), "MONO" (show 1 meter, an average of left and right), "MONOL" (show 1 meter, taking the LEFT audio level info), or "MONOR" (as MONOL, but take the RIGHT audio level info).

AudioOffset is again optional. By default the meters show audio in the range of -40dbFS to -12dbFS (so -22dbFS as 0dbu with the top level of -12 showing +10dbu peak level (PPM 6.5). The Offset is used to adjust if the audio card isn't lined up to this. For example applying a setting of -10 will move the range to -50dbFS to -22 dbFS. This is applied on the local meter and if the Broadcast setting is enabled, the

adjusted value is sent to client VMeters. It is possible to also specify a DBOffset on a client to further adjust this level, but usually a client would be set to 0 and the adjustment would be made at the source.

SilenceParameters is where you specify the dbFS level below which silence is detected, and the duration that it has to be silent for before. Default is -75dbFS for 10 seconds. Range is -100 (ie never detect silence) to 0. Format is "<level>,<durationInSeconds>".

EmailSilenceTo, EmailSilenceCc and EmailSilenceBcc are comma or semi-colon separated list of email addresses to send the silence alerts to. Default is blank. Emails will only be sent if the email settings are defined under the [**Notifications**] section. It is also possible to set some global email addresses in the [**Notifications**] section, so these fields can be left blank unless you have specific people that you want to email for each meter separately.

EmailLostConnectionTo, **EmailLostConnectionCc** and **EmailLostConnectionBcc** are similar fields for lost connection alerts.

EmailSilenceStartSubject, **EmailSilenceEndSubject**, **EmailLostConnectionStartSubject**, and **EmailLostConnectionEndSubject** are the subject of the email sent for each situation (start meaning when that alert is triggered and end meaning when it is returned to a normal state). In each case, %NAME% can be used as a variable to insert the meter name, as defined in the **Name** setting. Setting this will override the global setting in the Notifications section.

EmailSilenceStartExtraDetail, **EmailSilenceEndExtraDetail**, **EmailLostConnectionStartExtraDetail**, and **EmailLostConnectionEndExtraDetail** are text strings that can be added to the relevant emails when sent. They could be used to give more detail on the audio feed, for example. Setting this will override the global setting in the Notifications section.

PostLocation is the URL to do a HTTP Post to in the event of silence or lost connection alerts. It is possible to set this in the **[Notifications]** section as a global setting. This **[MeterX]** setting overrides it.

PostSilenceStart and **PostSilenceEnd** are the data of what is posted on silence (ie the part after the "?"), and **PostConnectionLost** and **PostConnectionRestored** are what is posted when connection is lost to a master VMeter / Livewire / Wheatstone port.

FileLocation allows you to specify a file to write log entries to. Setting this under each Meter overrides the settings in the Notifications section (see below). The FileLocation can contain variables to specify the filename, for example:

FileLocation=C:\LOGS\%year-%month.LOG

FileFormat specifies what to actually write in the file. Again this uses variables. For example:

FileFormat= %day/%month/%year %hour:%min:%sec:\t%METER\t%CHANNEL\t%STATE\t%DURATION

\t means tab, so the above file would open nicely in Excel.

Hidden allows you to hide this particular meter. All other aspects will function as normal (reading the levels, broadcasting to other meters, silence detection/notification and so on). Default is False.

The **[Notifications]** section defines the server properties to use whenever sending emails:

```
[Notifications]
FromName=%CHANNELNAME% Silence Detector (VMeters)
FromAddress=user@gmail.com
SMTPServer=smtp.gmail.com
SMTPUser=user@gmail.com
SMTPPassword=MyPassword
SMTPTimeout=20
TLSPort=587
EmailSilenceTo=cjvoce@gmail.com
EmailSilenceCc=
EmailSilenceBcc=
EmailLostConnectionTo=cjvoce@gmail.com
EmailLostConnectionCc=
EmailLostConnectionBcc=
PostLocation=http://localhost:18513/VClock
EmailSilenceStartSubject=VMeters - Silence detected on %NAME%
EmailSilenceEndSubject=VMeters - Audio returned on %NAME%
EmailSilenceStartExtraDetail=
EmailSilenceEndExtraDetail=
EmailLostConnectionStartSubject=VMeters - Lost Connection detected on %NAME%
EmailLostConnectionEndSubject=VMeters - Connection re-established on %NAME%
EmailLostConnectionStartExtraDetail=
EmailLostConnectionEndExtraDetail=
FileLocation=
FileFormat=
```

FromName is the “pretty” name shown in most email clients. **%CHANNELNAME%** will be replaced with the name of the meter as defined in the **[MeterX]** section.

FromAddress is the actual email address it is sent from. Note that with some servers (Gmail for example) this must match the user you are logging in as.

SMTPServer specifies the Server to use

SMTPUser and **SMTPPassword** are used to authenticate to the Server if required. Leave blank if not.

SMTPTimeout is how long to wait before giving up sending the email. Default 20 seconds is usually enough.

TLSPort is used mainly for Gmail (set it to 587 for Gmail). Leave as 0 to use standard SMTP Server.

EmailSilenceTo, **EmailSilenceCc** and **EmailSilenceBcc** are comma or semi-colon lists of global email recipients if silence occurs.

EmailLostConnectionTo/Cc/Bcc are the email recipients when a client loses its connection / re-establishes its connection with its server.

EmailSilenceStartSubject, **EmailSilenceEndSubject**, **EmailLostConnectionStartSubject**, and **EmailLostConnectionEndSubject** are the subject of the email sent for each situation (start meaning when that alert is triggered and end meaning when it is returned to a normal state). In each case, %NAME% can be used as a variable to insert the meter name, as defined in the **Name** setting. This can be overridden in the Meters section for each meter.

EmailSilenceStartExtraDetail, **EmailSilenceEndExtraDetail**, **EmailLostConnectionStartExtraDetail**, and **EmailLostConnectionEndExtraDetail** are text strings that can be added to the

relevant emails when sent. This can be overridden in the Meters section for each meter.

PostLocation is a global setting for the http server to post to on silence/connection loss. The **[MeterX] PostLocation** setting overrides this if set.

FileLocation allows you to globally specify a file to write log entries to. Setting this under each Meter section (see above) overrides the settings in this Notifications section. The FileLocation can contain variables to specify the filename, for example:

FileLocation=C:\LOGS\%year-%month.LOG

FileFormat specifies what to actually write in the file. Again this uses variables. For example:

FileFormat= %day/%month/%year %hour:%min:%sec:\t%METER\t%CHANNEL\t%STATE\t%DURATION

\t means tab, so the above file would open nicely in Excel.