Alsa Opensrc Org

Independent ALSA and linux audio support site

How to use softvol to control the master volume

From the ALSA wiki

Jump to: navigation, search

This howto describes a workaround if your master volume doesn't work. This happens if your sound card can't control the volume on the hardware side or the driver doesn't support this feature of your sound card. Maybe updating ALSA or using another module will fix the problem. If nothing works, you can define a new virtual pcm device in the .asoundrc file, which controls the volume on the software side.

Contents

Preparations

Find out on which existing PCM device you can base your setup. In this device, the audio data will be processed the last on its way to the sound card. In a simple stereo setup, this is problably just the `hw:0,0` device. If your card doesn't support hardware mixing, you may have to use a `dmix` plugin first (see example below). In a typical 5.1 surround sound setup, you are probably using the predefined `surround51` device.

To get a list of possible devices, you may use:

Alsa Opensrc Org

To test this device, use:

```
speaker-test -D<device name> -c<channel count> -twav
```

If that command produces sound on the correct channels and you can use it on two different consoles simultaneously, you can use this device. If simultaneous usage doesn't work, see dmix and Hardware mixing, software mixing to enable software mixing.

Editing the asoundrc file

Creating a new softvol device

Open the asoundrc file in your favorite editor. E.g. like this:

```
nano ~/.asoundrc
```

Now we create a new softvol device be typing:

```
pcm.softvol {
    type            softvol
    slave {
        pcm         "<device name>"
    }
    control {
        name        "<control name>"
        card        0
    }
}
```

This will create a new PCM device called `softvol`, which is controlled by a volume control `<control name>` and which will pass the sound data with the changed volume to its *slave* `<device name>`.

You have to replace `<device name>` with the name of the device you determined above and `<control name>` with what you want to call your new volume control, e.g. `SoftMaster`. If your card does not have a master volume control at all, you're lucky, because you can name your new volume control `Master` and your new control works like a master volume control is supposed to. To find out, whether such a control exists, run:

```
amixer controls | grep Master
```

If this command lists a control named `Master`, you should not name your new control like this. Unfortunately, existing controls can't be overwritten, so you have to pick a name like `SoftMaster`. This control will now control everything, but as it is not called `Master`, mixers (like KMix) won't use it to control master volume, unless you can make them choose another control (like GMix).

The name you give to your control matters a lot. Some suffixes have special meanings. For example, if you want your softvol to control the playback volume only, the control name must end with `Playback Volume`. Such a name prevents the mixer from showing it as a capture control.

Now test your new device with:

```
speaker-test -Dsoftvol -c<channel count> -twav
```

Note: The new volume control won't appear immediately! Only after the first usage of the newly defined device (e.g. with the command above), should `amixer controls | grep <control name>` display your new control. Mixers that were already started before the first usage (like KMix) have to be restarted to adopt the changes. If you still don't see the new control, try restarting ALSA or your PC.

Make applications use it

Finally, we'll have to make all applications use this new device. In a simple stereo setup, we can redefine the default device and route it to our `softvol` device (with a `plug` device, so rate is converted automatically). In that case, add this to your asoundrc file:

```
pcm.!default {
    type            plug
```

Alsa Opensrc Org

With this configuration, our `softvol` device controls both playback and capture. This may not work properly for some setups. If you prefer that `softvol` controls the playback only, you must define a new default device which is of type `asym` : you can then decide that the playback is controlled by the softvol, and let the capture unchanged. In that case, you should add this to your asoundrc file:

```
pcm.!default {
    type            asym
    playback.pcm {
        type        plug
        slave.pcm   "softvol"
    }
    capture.pcm {
        type        plug
        slave.pcm   "<device name>"
    }
}
```

If you have a multi channel sound card, you may want to upmix these stereo signals first (see SurroundSound). It is useful to redefine the `surround40` , `surround51` ... devices in the same way, so everything is passed through our new softvol device by default. **Note that you should not overwrite the device `<device name>` from above!**

Make sure that every application uses a device that is redirected to your softvol device because everything else will not be controlled and may be too loud! If you can't redefine the default devices, you have to configure your applications separately.

Note, if your `<device name>` happened to be named "**default**" literally, you will have to go back to the first step, and use "**cards.pcm.default**" instead of just "**default**" in pcm.softvol slave pcm block. Otherwise, when trying to replace default output, you will get error

```
ALSA lib conf.c:4049:(snd1_config_check_hop) Too many definition levels
Playback open error: -22,Invalid argument
```

Common example with dmix

In the latest ALSA versions (after 1.0.9) dmix has been enabled by default for the boards tha ... th features (softvol+dmix) you must have in \~/.asoundrc something like this:

```
pcm.!default {
    type            plug
    slave.pcm       "softvol"   #make use of softvol
}

pcm.softvol {
    type            softvol
    slave {
        pcm         "dmix"      #redirect the output to dmix (instead o
    }
    control {
        name        "PCM"       #override the PCM slider to set the sof
        card        0
    }
}
```

In this case, the device called `dmix` is the device `<device name>` the whole setup is based on (see above).

This works for my crappy C-Media Electronics CMI 9739 - nforce2 integrated 'soundcard' that lacks both volume control and mixing in hardware. I think it will do for many other similar 'soundcards'.

More complex example

I am using an *SBLive! Platinum [CT4760P]* and the asoundrc file below. Maybe you can solve your problems by understanding this example and maybe copy parts of it.

On the lowest level, I have two `softvol` devices that pass their data to the predefined devices `front` and `rear` controlling their volume with the controls `Front Master` and `Rear Master` . A `multi` plugin merges those two stereo devices into a four channel device. My `multi` device would be the `<device name>` device in the text above. The device called `softvol` controls the volume with a control called `SoftMaster` using the `multi` device as slave. I then define an upmix device to upmix stereo streams to 4.0 and some downmix devices to downmix 4.1, 5.0, 5.1 and 7.1 streams to 4.0.

To enable recording with multiple applications, I define some `dsnoop` devices. `dsnoop` does the same thing with recording as `dmix` does with playback. The device `recording` is a regular stereo recording device, whereas `recleft` and `recright` are mono devices recording only one channel of the stereo stream. If you want to plug two mono mics into the stereo mic plug of your sound card (with an adapter) and record from them separately, this is quite handy, otherwise, this part is not necessary.

Finally I replace the `default` device with a `asym` device, redirecting its playback to the upmixing device and its recording to the recording device. This way, the `default` device is playback and recording device at the same time (full duplex). I also create the `surroundX` devices redirecting to the corresponding downmix devices.

What I didn't consider yet in my file are devices needed for compatibility with OSS and similar. If I need them one day and change my config file locally, I'll post an update here.

```
#-------------------------------
#  Volume
#-------------------------------

# volume of all channels
pcm.softvol {
    type        softvol
    slave.pcm   "multi"
    control {
        name    "SoftMaster"
        card    0
    }
}

# splitting the channels in front and rear
pcm.multi {
    type    multi
    slaves {
        a.pcm        "frontvol"
        a.channels   2
        b.pcm        "rearvol"
        b.channels   2
    }
    bindings {
        0.slave      a
```

```
        0.channel     0
        1.channel     1
        2.slave       b
        2.channel     0
        3.slave       b
        3.channel     1
    }
}

# front
pcm.rearvol {
    type        softvol
    slave.pcm   "rear"
    control {
        name    "Rear Master"
        card    0
    }
}

# rear
pcm.frontvol {
    type        softvol
    slave.pcm   "front"
    control {
        name    "Front Master"
        card    0
    }
}

#-------------------------------
#  Recording
#-------------------------------

pcm.recording {
    type        dsnoop
    ipc_key     2589
    slave {
        pcm     "hw:0,0"
        format  "S16_LE"
    }
```

```
}

pcm.recleft {
    type        dsnoop
    ipc_key     2589
    slave {
        pcm     "hw:0,0"
        format  "S16_LE"
    }
    bindings.0 0
}

pcm.recright {
    type        dsnoop
    ipc_key     2589
    slave {
        pcm     "hw:0,0"
        format  "S16_LE"
    }
    bindings.0 1
}

#-------------------------------
#  Upmix
#-------------------------------

# upmix stereo to 40
pcm.upmix {
    type        route
    slave.pcm   "softvol"
    slave.channels    4
    ttable {
        0.0    1
        0.2    1
        1.1    1
        1.3    1
    }
}

#-------------------------------
#  Downmix
```

```
#---------------------------------

pcm.downmix41 {
    type        route
    slave.pcm   "softvol"
    slave.channels    4
    ttable {
        0.0    1
        1.1    1
        2.2    1
        3.3    1
    }
}

pcm.downmix51 {
    type        route
    slave.pcm   "softvol"
    slave.channels    4
    ttable {
        0.0    0.67
        1.1    0.67
        2.2    1
        3.3    1
        4.0    0.33
        4.1    0.33
    }
}

pcm.downmix71 {
    type        route
    slave.pcm    "softvol"
    slave.channels    4
    ttable {
        0.0    0.34
        1.1    0.34
        2.2    0.67
        3.3    0.67
        4.0    0.33
        4.1    0.33
        6.0    0.33
        6.2    0.33
```

```
    }
}

#-------------------------------
#  Overwrite existing devices
#-------------------------------

pcm.!default {
    type           asym
    playback.pcm   "plug:upmix"
    capture.pcm    "plug:recording"
}

pcm.!surround40 {
    type         plug
    slave.pcm    "softvol"
}

pcm.!surround41 {
    type         plug
    slave.pcm    "downmix41"
}

pcm.!surround50 {
    type         plug
    slave.pcm    "downmix51"
}

pcm.!surround51 {
    type        plug
    slave.pcm    "downmix51"
}

pcm.!surround71 {
    type         plug
    slave.pcm    "downmix71"
}
```

See also

- asoundrc
- SurroundSound
- dmix

Retrieved from "http://alsa.opensrc.org/How_to_use_softvol_to_control_the_master_volume"

Category: Howto

GITHUB | EDIT