OBS Basics

Open Broadcast Software (kurz OBS) ist ein Open-Source-Programm, welches eine einfache und freie Möglichkeit gibt, vom lokalen Computer zu unterschiedlichen Anbietern zu streamen. OBS kann für verschiedene Betriebssysteme heruntergeladen und erweitert werden. Die Beliebtheit ist der einfachen Handhabung, sowie der vielseitigen Einsatzmöglichkeit zu verdanken, weshalb die Gaming- und Twitch Community großflächig dieses Tool einsetzt.

Download und Installation

OBS steht für verschiedene Betriebssystemen zur Verfügung und kann von der offiziellen Seite, sowie aus den Paketquellen des jeweiligen Paketmanagers heruntergeladen werden.

Windows

Installer herunterladen und ausführen:

OBS Studio Website: https://obsproject.com/download

GitHub Releases: https://github.com/obsproject/obs-studio/releases

Installation via den Microsoft Store:

Microsoft Store: [click]

Installation via Steam:

Steam:[click]

Aus dem Quellcode kompilieren

Dokumentation: https://obsproject.com/wiki/build-instructions-for-windows

macOS

details

Disk Image (.dmg) herunterladen und die OBS.app in den Programme Ordner schieben

OBS Studio Website: https://obsproject.com/download

GitHub Releases: https://github.com/obsproject/obs-studio/releases

Installation via Steam:

Steam:[click]

Aus dem Quellcode kompilieren

Dokumentation: https://obsproject.com/wiki/build-instructions-for-mac

GNU/Linux

details

Die Installation für Linux kann auf verschiedenen Wegen erfolgen.

Flatpak: Installation via Flathub

Ubuntu:

```
sudo add-apt-repository ppa:obsproject/obs-studio
        sudo apt update
        sudo apt install obs-studio
```

Arch Linux und Manjaro:

details

```
sudo pacman -S obs-studio
```

Im AUR befinden sich verschiedene PKBUILDs um OBS mit unterschiedlichen Modulen zu bauen:

- obs-studio-tytan652
  - OBS mit VST 2 Plugin, RIST Protokol und AJA Device Support.
  - Integrierten Browser Modul
  - Kleine Anpassungen um das Leben leichter zu machen
- obs-studio-browser

    - OBS mit Integrierten Browser Modul und VST
- obs-studio-rc
    - Release Canidate und Beta Version

Debian:

details

```
sudo apt update
        sudo apt install ffmpeg obs-studio
```

Aus dem Quellcode kompilieren

Dokumentation: https://obsproject.com/wiki/build-instructions-for-linux

Fedora:

details

```
# RPM Fusion hinzufügen, falls noch nicht geschen
        sudo dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E  %fedora).noarch.rpm https://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-$(rpm -E %fedora).noarch.rpm
        # Installation OBS
        sudo dnf install obs-studio
```

Gentoo:

details

```
sudo emerge media-video/obs-studio
```

NixOS:

details

```
nix-env -i obs-studio
```

OpenMandriva Lx4

details

```
dnf install obs-studio
```

Solus

details

```
eopkg install obs-studio
```

Void

details

```
sudo xbps-install obs
```

FreeBSD

details

Die Installation für FreeBSD erfolgt via `pkg`.

```
pkg install obs-studio
```

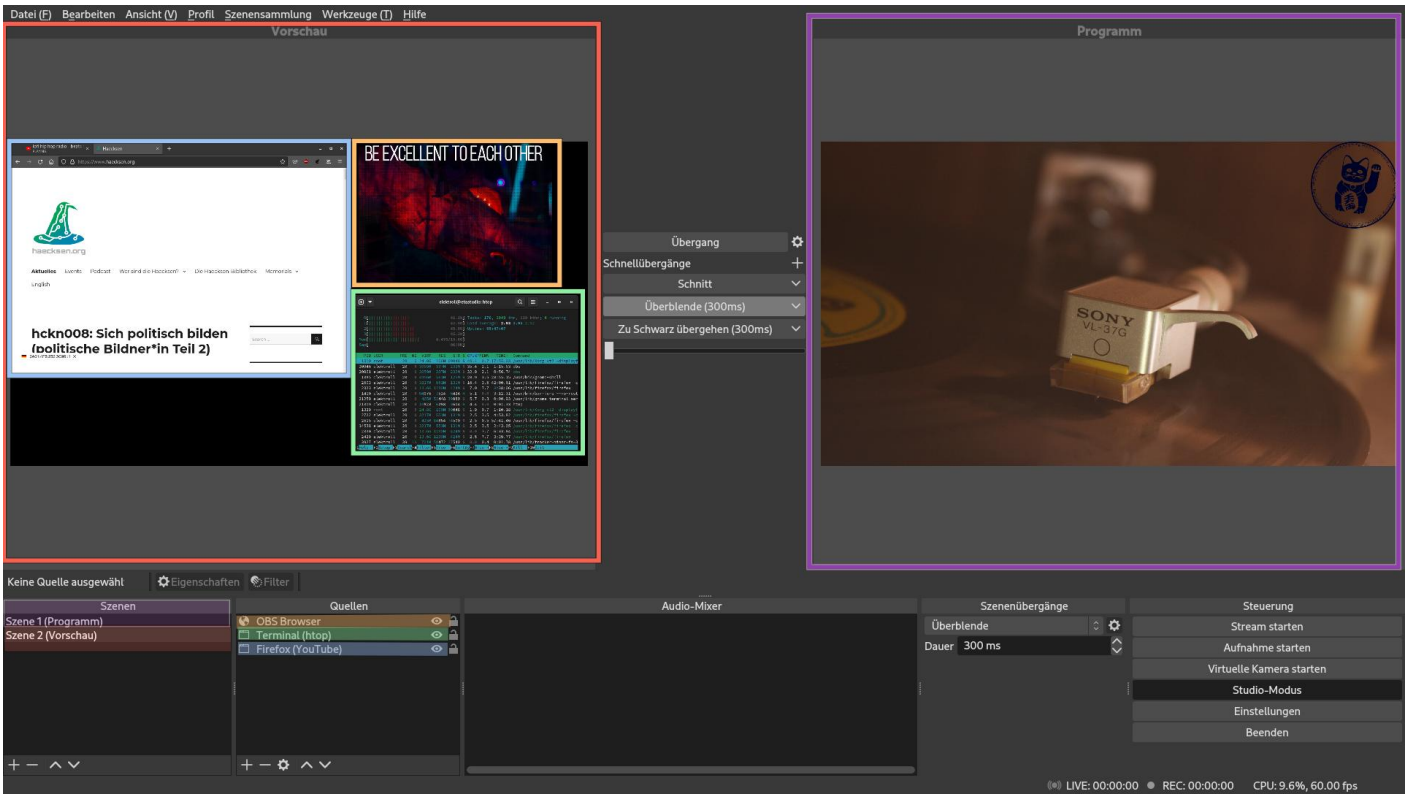
OpenBSD

details

Für OpenBSD besteht ein Work In Progress Port auf GitHub unter:

https://github.com/jasperla/openbsd-wip

Einführung

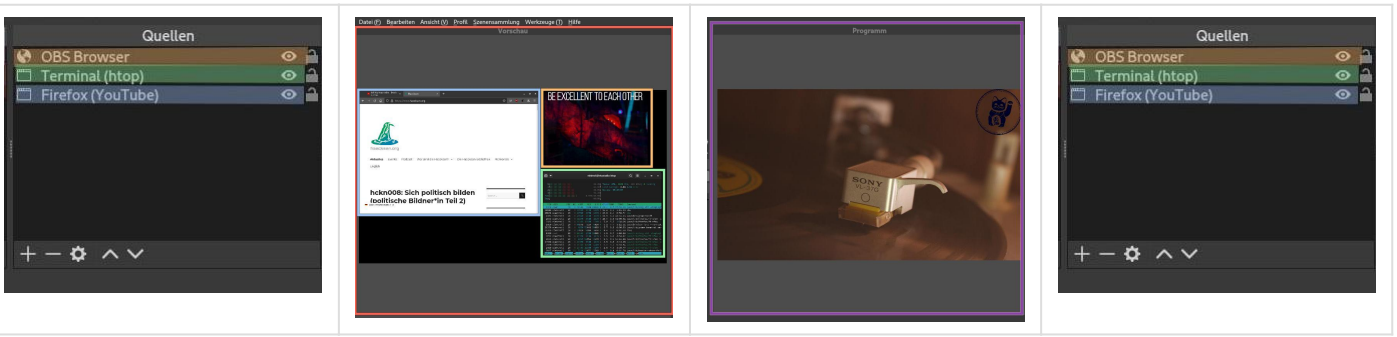


Screenshot OBS Version 27.1.3 unter Arch Linux, kompiliert mit Browser Modul.

In dieser Dokumentation kann das Äußere/Aussehen von OBS je nach Betriebssystem und Version leicht variieren. In der Standardansicht von OBS, wird nur die aktuell laufende Szene angezeigt. Es ist zu empfehlen, unten rechts auf den **Studio-Modus** zu wechseln, in dem man die gleichnamige Schaltfläche dafür anklickt.

Im Studio-Modus bietet OBS eine getrennte Ansicht, zwischen dem was aktuell gesendet wird (rechts) und dem was in Vorbereitung (links) ist.

Dem Screenshot kann man verschiedene farbliche Markierungen entnehmen. Das linke Fenster in **Orange** entspricht der Szene mit dem Namen **"Szene 2 (Vorschau)"**. Das rechte Fenster in **Lila** entspricht somit der Szene **"Szene 1 (Programm)**". Die Benennung der Szenen kann frei gesetzt werden. Das ist möglich über einen Rechtsklick auf die Szene und dann die Schaltfläche "Umbennen"/"Rename".

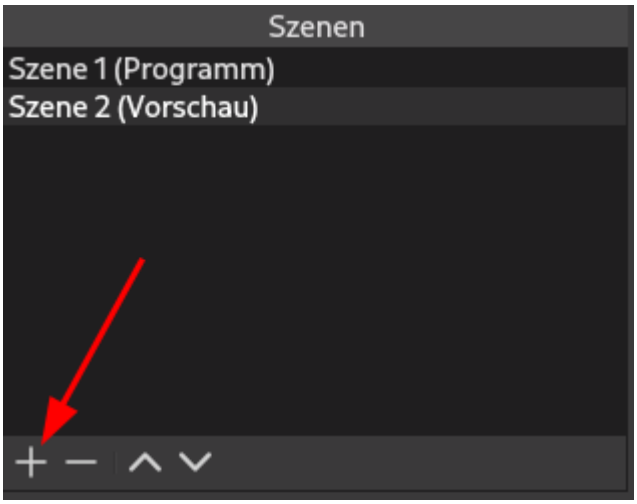


In einer Szene können mehrere Quellen hinterlegt werden. Im linken Vorschaufenster (Orange) sind drei unterschiedliche Quellen zusehen, welche gerade vom Bildschirm abgegriffen werden und frei positioniert wurden. Diese Quellen sind als Ebenen übereinander gelegt. Dies kann man sich wie Klarsichtfolien vorstellen, welche jeweils ein unterschiedliches Motiv zeigen und übereinander gelegt sind. Zu einem haben wir ganz oben das gelbliche Fenster namens OBS Browser. Hierbei handelt es sich um einen integrierten Browser im OBS, welcher eine URL aufruft und die Seite rendert. Darunter in Grün wird ein Terminal mit dem Taskmanager htop aufgenommen und zum Schluss wurde in diesem Beispiel noch ein Firefox Fenster hinzugefügt. Die Szenen können über einen Klick gewechselt werden, sodass Szene 2 Live geht.

Neue Szene hinzufügen

Um eine eigene neue **Szene** hinzuzufügen, muss lediglich im Szenen Fenster auf das **Plus** geklickt und der **Szene** ein Name gegeben werden. Hierbei wird eine neue und leere Szene erstellt, welche wir befüllen können. Zum Beispiel mit dem Abgreifen eines Fensters. Für dieses Beispiel benennen wir die neu erstellte Szene **"Szene 3"** und fügen im benachbarten Fenster eine neue **Quelle hinzu** .

> Beachtet das sich die Quellen im Namen unterscheiden können, je nach dem welches Betriebssystem benutzt wird und welcher grafischer Server im Einsatz ist. In diesem Beispiel wird unter Arch Linux mit X11 gearbeitet, weshalb die Fensteraufnahme den Beititel Xcomposite trägt.



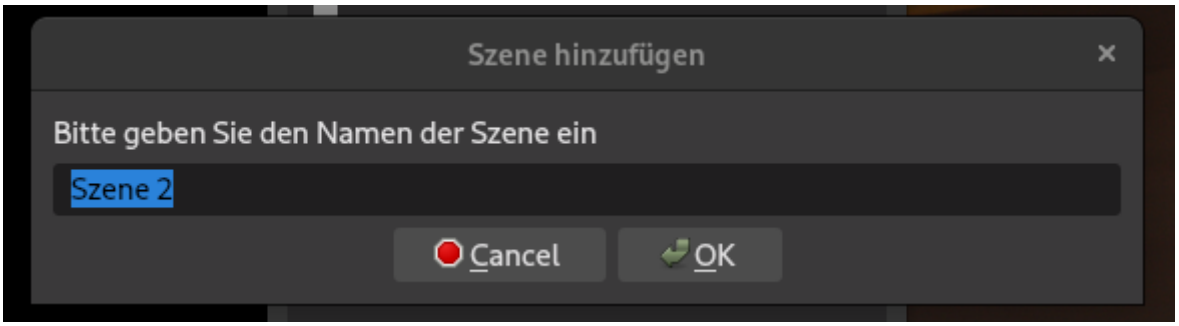
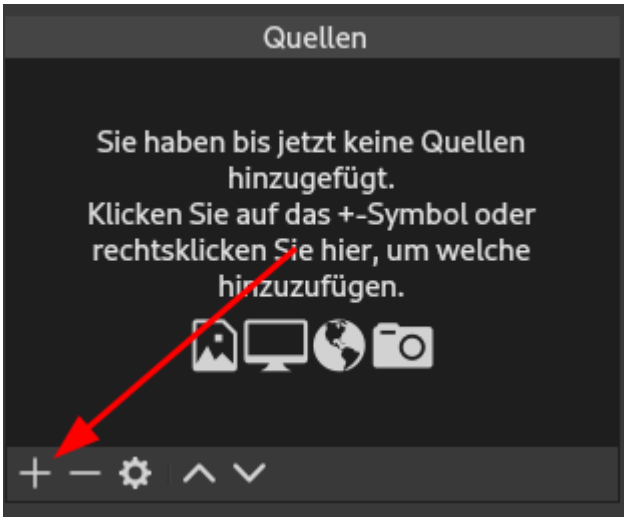
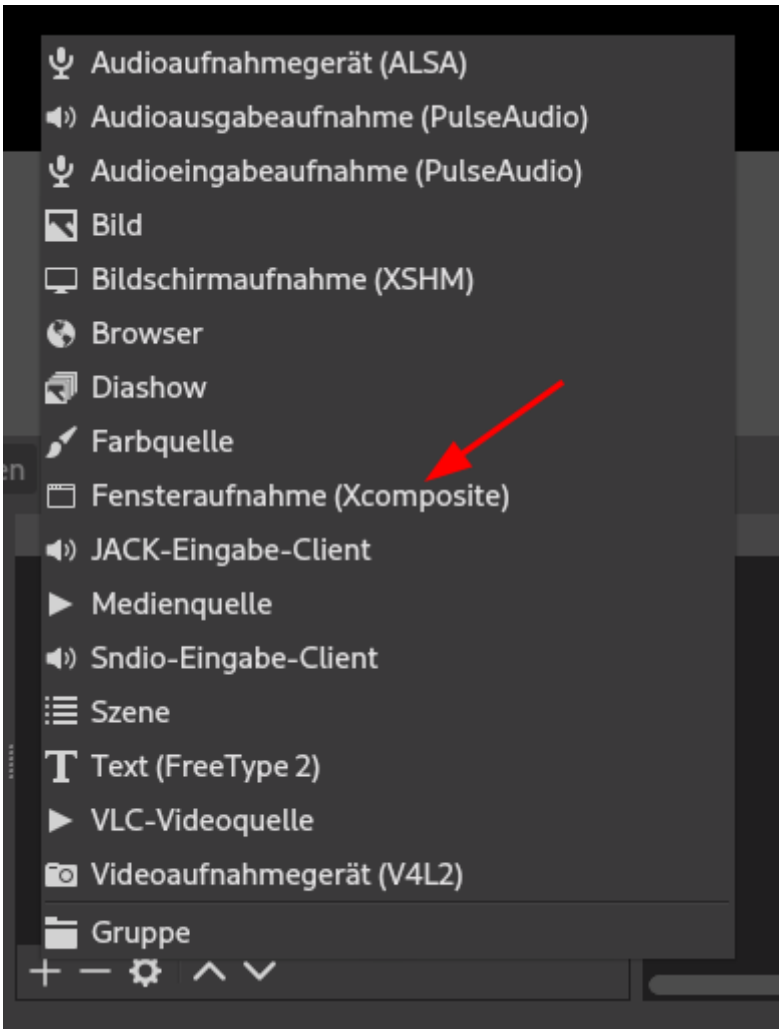


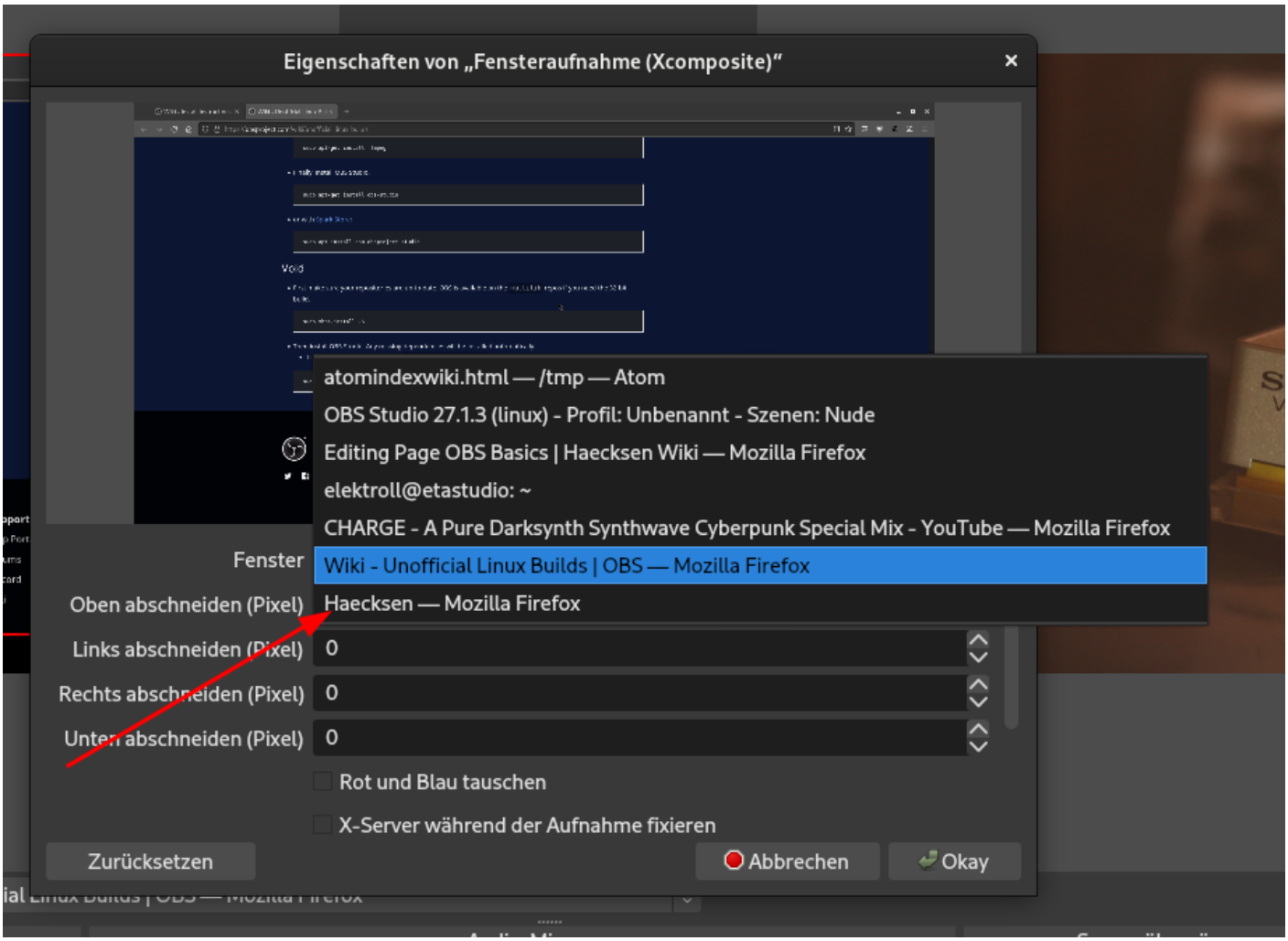
Bild Quellen hinzufügen

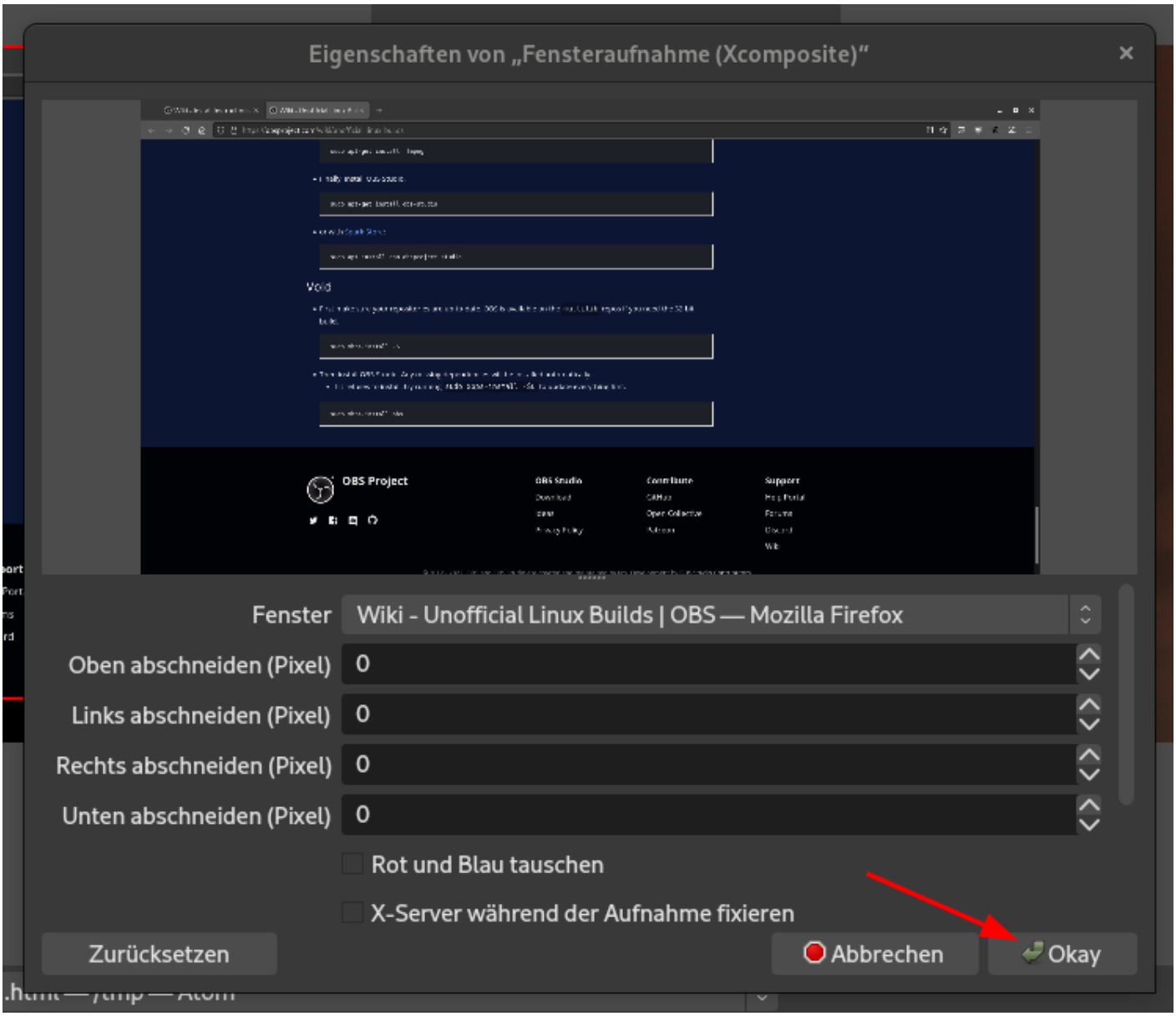
Da wir nun eine neue **Szene** haben, klicken wir ebenfalls auf das **Plus** Symbol bei den **Quellen** und fügen eine neue **Quelle** hinzu. Diese nennt sich **Fensteraufnahme**.



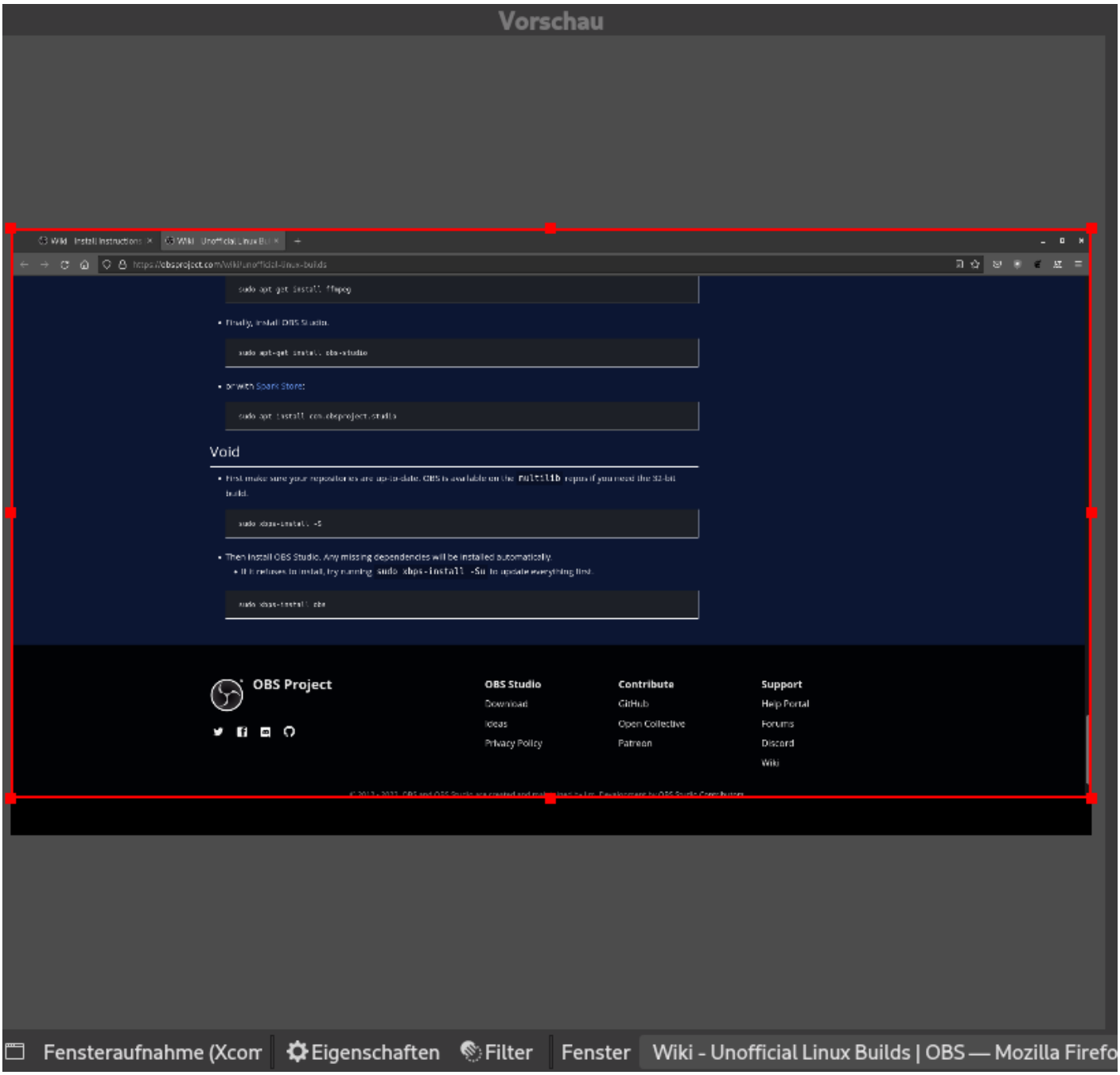


In dem darauf folgenden Fenster können wir das Fenster definieren, welches aufgezeichnet werden soll. Hierzu muss auf die Schaltfläche Fenster geklickt werden und das jeweilige Fenster ausgewählt werden. Daraufhin können noch vereinzelte Optionen angepasst werden. Bestätigt wird dies mit Okay.

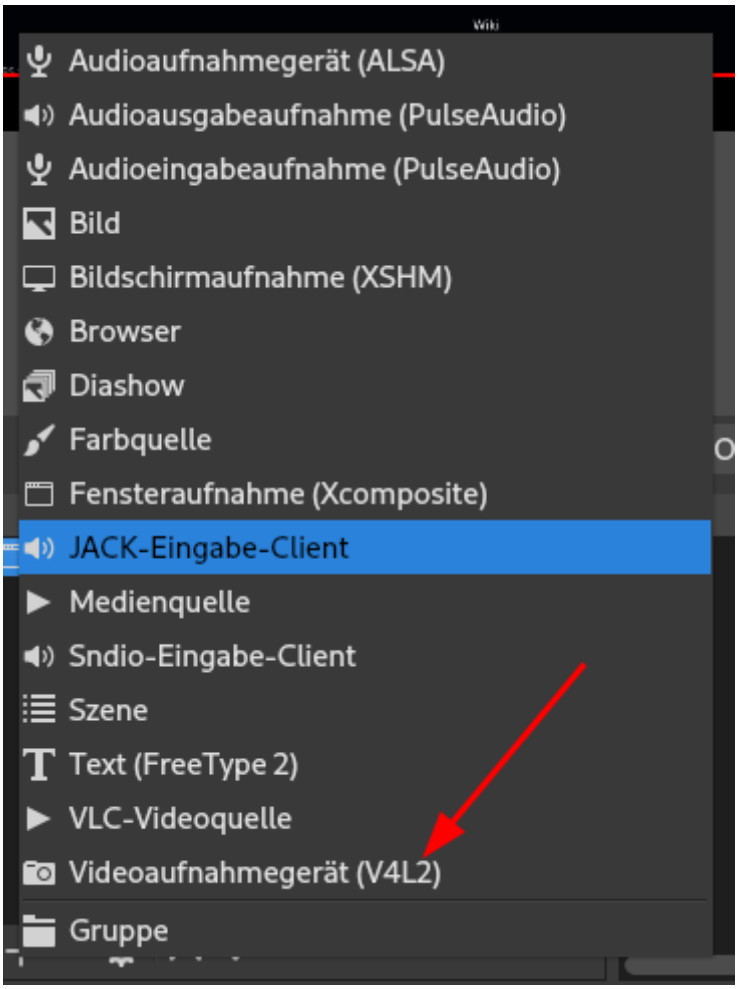

Eigenschaften von „Fensteraufnahme (Xcomposite)“
atomindexwiki.html — /tmp — Atom
OBS Studio 27.1.3 (linux) - Profil: Unbenannt - Szenen: Nude
Editing Page OBS Basics | Haecksen Wiki — Mozilla Firefox
elektroll@etastudio: ~
CHARGE - A Pure Darksynth Synthwave Cyberpunk Special Mix - YouTube — Mozilla Firefox
Wiki - Unofficial Linux Builds | OBS — Mozilla Firefox
Haecksen — Mozilla Firefox
Fenster
Oben abschneiden (Pixel)
Links abschneiden (Pixel)
0
Rechts abschneiden (Pixel)
0
Unten abschneiden (Pixel)
0
Rot und Blau tauschen
X-Server während der Aufnahme fixieren
Zurücksetzen
Abbrechen
Okay



Im Vorschaufenster sehen wir nun den Firefox Browser, welchen ich für diese Einführung ausgewählt habe. Die rote Markierung um das Fenster bietet mit die Möglichkeit das Fenster zu skalieren und somit zu verkleinern oder zu positionieren.



Zu der bestehenden Quelle fügen wir nun unsere Webcam als eine eigene Quelle hinzu. Dafür klicken wir erneut auf das Plussymbol und wählen **Videoaufnahmegerät**. (Das V4L2 steht für Video For Linux 2 und ist eine Linux spezifische Bezeichnung). In dem folgenden Fenster können noch Einstellungen durchgeführt und mit einem Klick auf "okay" bestätigt werden.


Audioaufnahmegerät (ALSA)
Audioausgabeaufnahme (PulseAudio)
Audioeingabeaufnahme (PulseAudio)
Bild
Bildschirmaufnahme (XSHM)
Browser
Diashow
Farbquelle
Fensteraufnahme (Xcomposite)
JACK-Eingabe-Client
Medienquelle
Sndio-Eingabe-Client
Szene
Text (FreeType 2)
VLC-Videoquelle
Videoaufnahmegerät (V4L2)
Gruppe

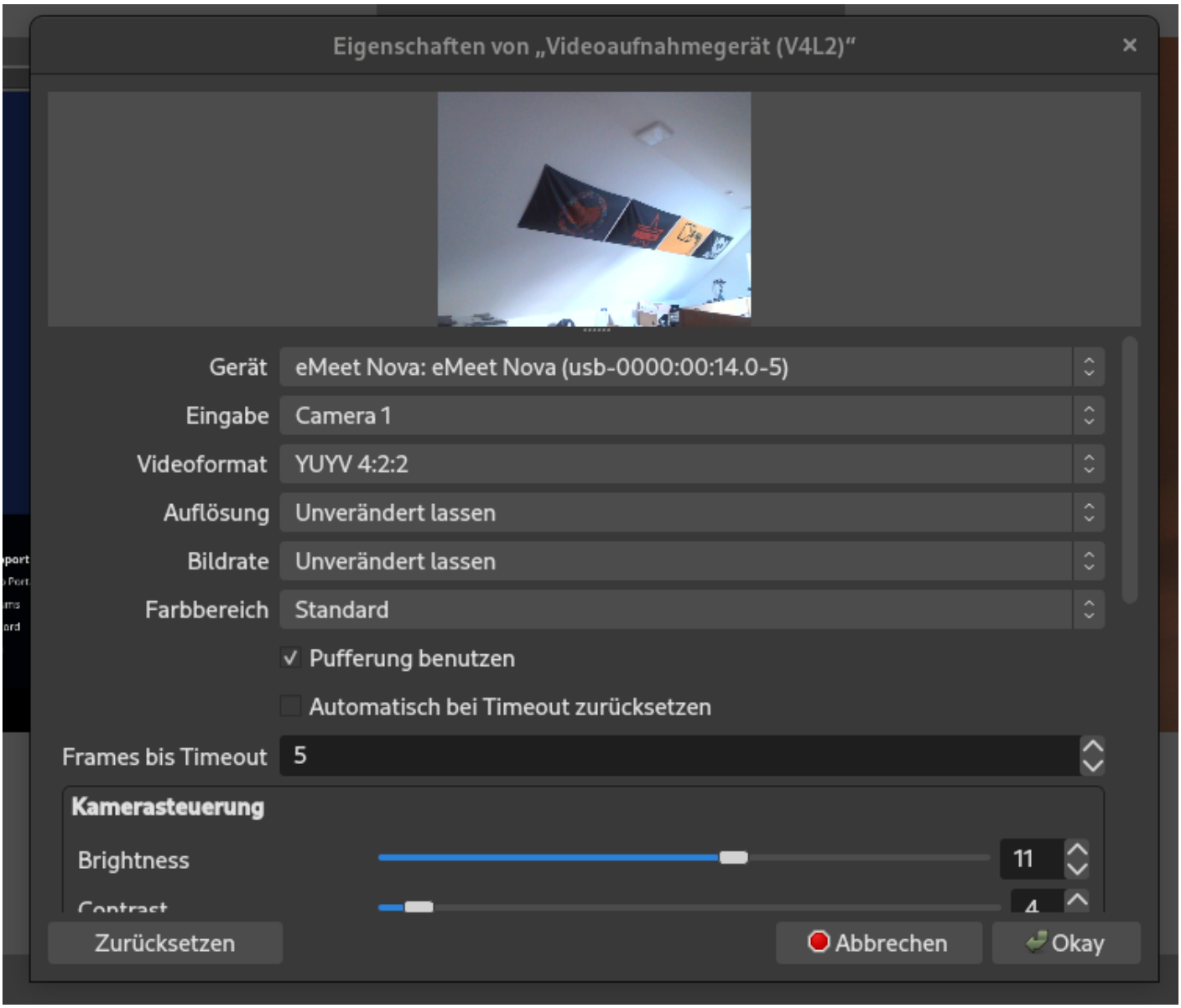
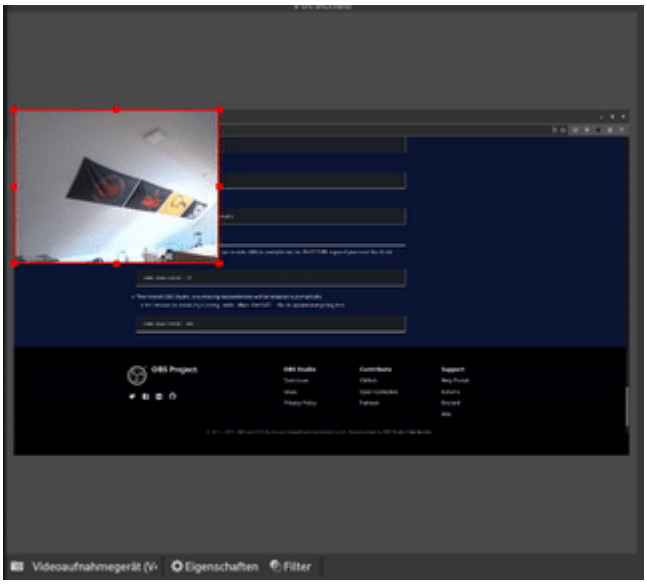


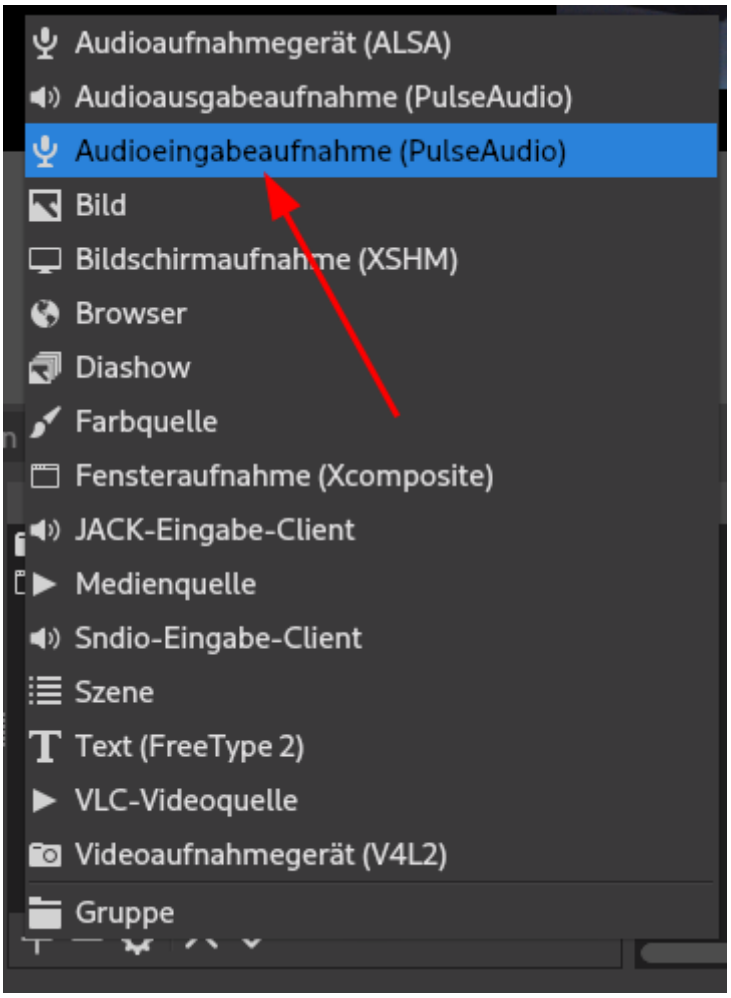
Bild Quellen anpassen

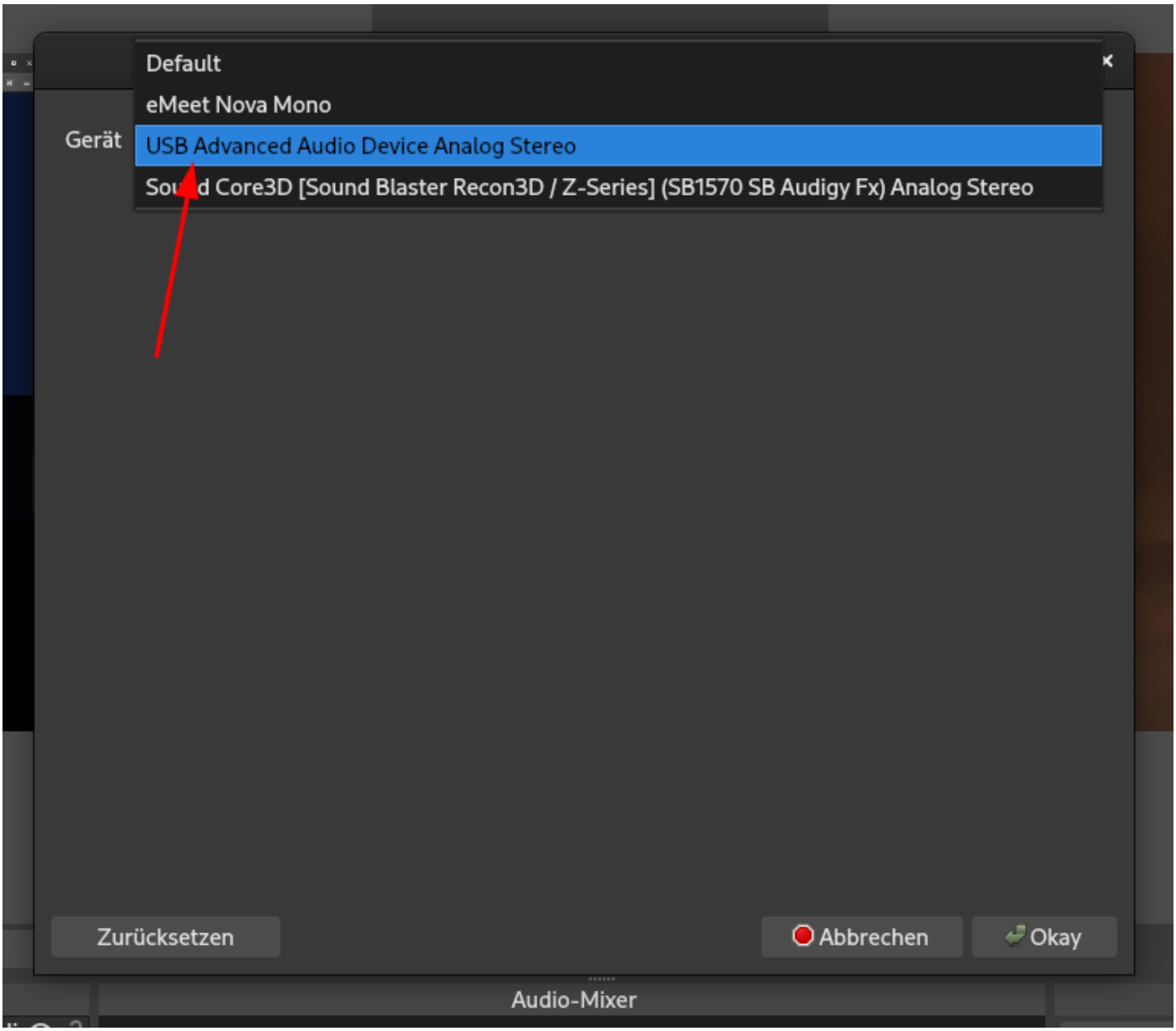
Es ist nun möglich beide Quellen im Vorschaufenster, via Drag-and-drop zu verschieben, zu skalieren und zuzuschneiden, indem wir die Objekte einfach anklicken und schieben. Für die Skalierung der Fenstergröße müssen die Ecken ausgewählt werden. **Wenn man dabei die ALT Taste festhält, wird das Fenster zugeschnitten.**



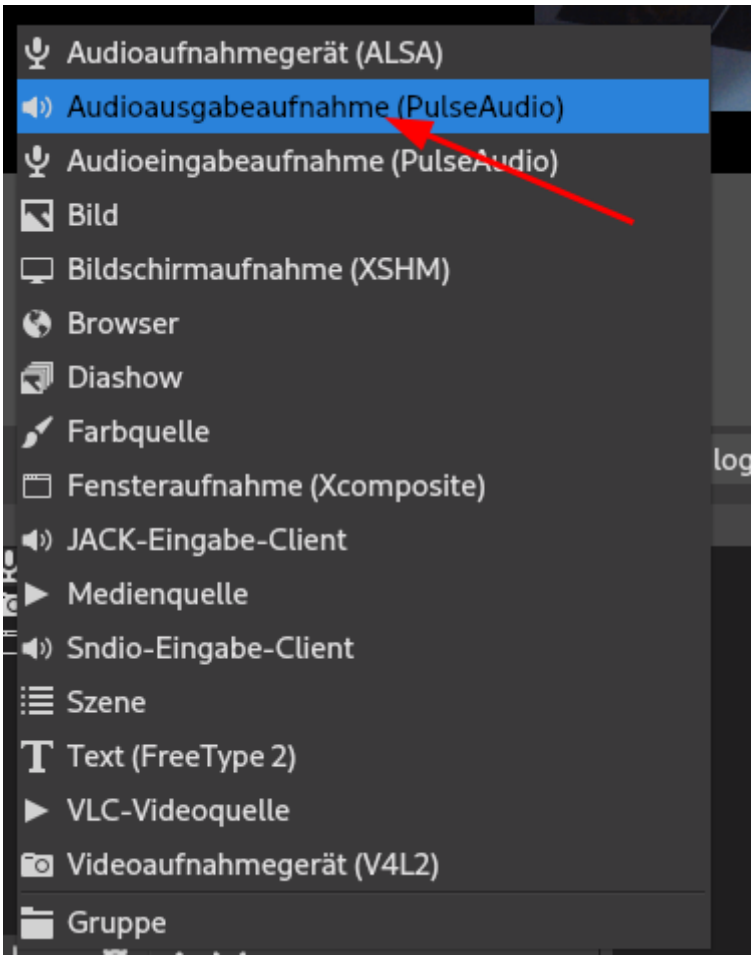
Audio Quellen

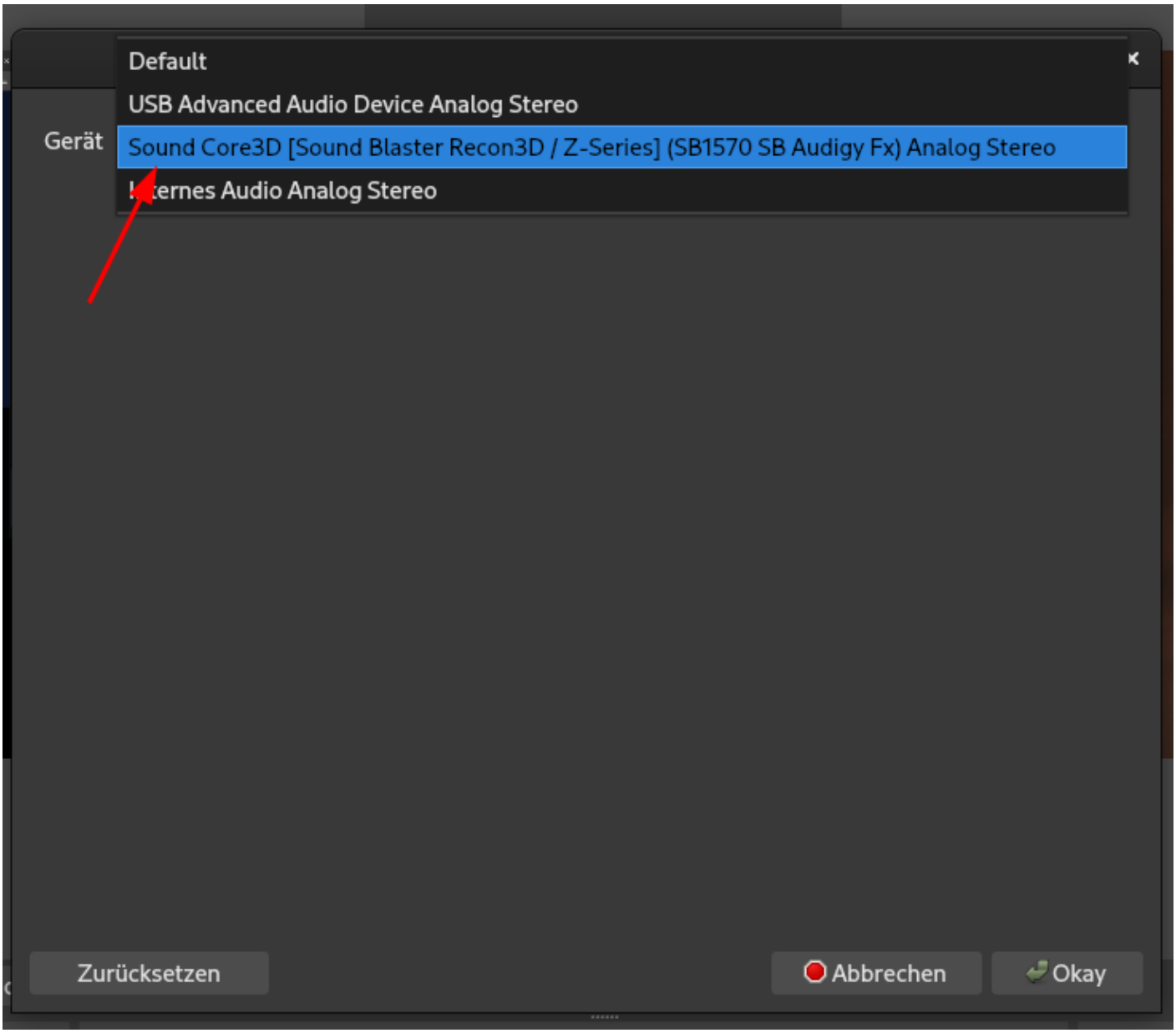
Wir haben nun die Bild-Quellen, welche wir aufnehmen wollen. Jetzt fehlen nur noch die Audio-Quellen wie **Desktop Audio** und ein **Mikrofon**. Diese werden wie die Bild-Quellen innerhalb der Szene als **Quelle** hinzugefügt. Um ein lokal angeschlossenes **Mikrofon** aufzunehmen, wählt man die **Audioeingabeaufnahme** (das PulseAudio steht für ein Linux spezifischen Audio-Server). In dem darauf folgenden Fenster kann das Mikrofon ausgewählten werden, welches als Audioeingabegerät genutzt werden soll. Bestätigt mit Okay.


Audioaufnahmegerät (ALSA)
Audioausgabeaufnahme (PulseAudio)
Audioeingabeaufnahme (PulseAudio)
Bild
Bildschirmaufnahme (XSHM)
Browser
Diashow
Farbquelle
Fensteraufnahme (Xcomposite)
JACK-Eingabe-Client
Medienquelle
Sndio-Eingabe-Client
Szene
Text (FreeType 2)
VLC-Videoquelle
Videoaufnahmegerät (V4L2)
Gruppe

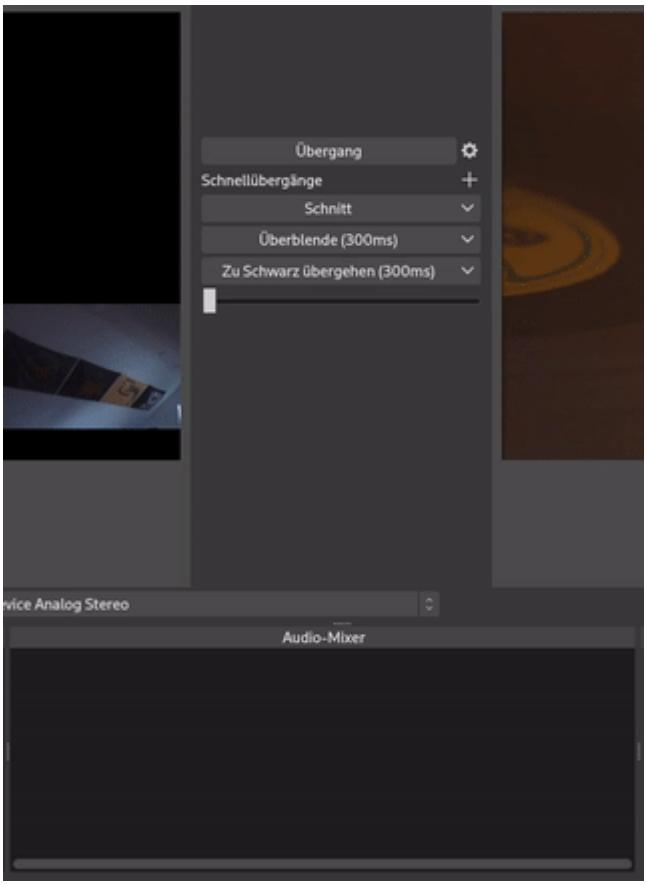


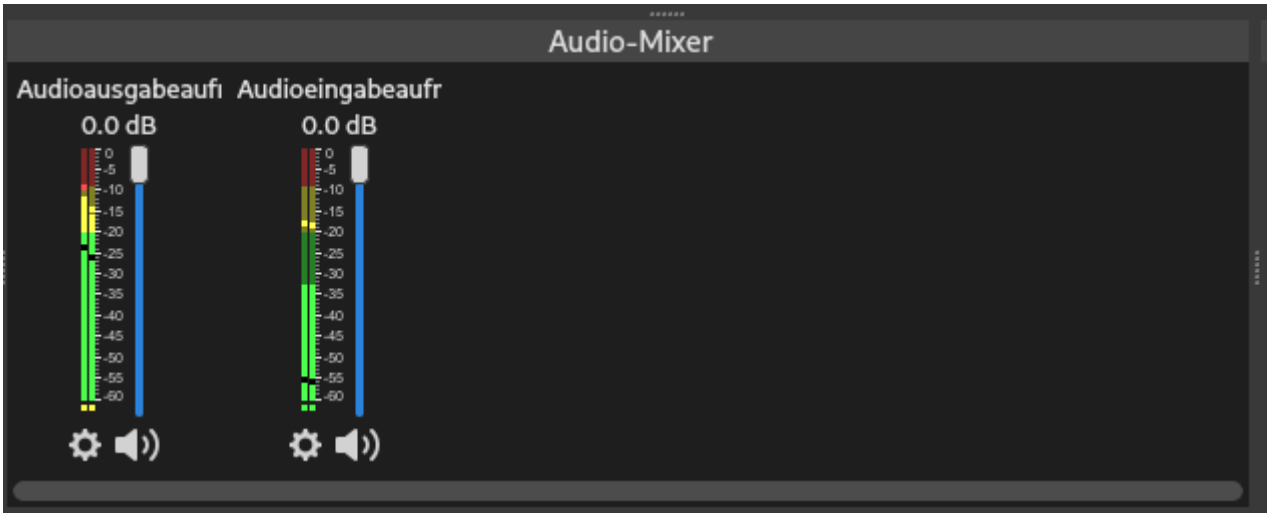
Wir wollen noch jenes Audio mitsenden, welches auf dem PC abgespielt wird. Sei es ein Spiel, Hintergrundmusik von einem Musikplayer oder ein Soundboard. Hierzu gehen wir genau so vor, wie oben beschrieben und wählen anstelle der Audioeingabeaufnahme nun **Audioausgabeaufnahme.** Wählt die Soundkarte aus, von welcher ihr das Audio abgreifen wollt. Die Gerätebezeichnungen werden mit sehr hoher Wahrscheinlichkeit andere sein, da in diesem Fall eine interne Soundkarte, eine zusätzliche Soundkarte und das Mikrofon als Soundkarte zur Auswahl stehen. Solltet ihr euch nicht sicher sein, ist die Default-Soundkarte meistens die richtige Wahl.


Audioaufnahmegerät (ALSA)
Audioausgabeaufnahme (PulseAudio)
Audioeingabeaufnahme (PulseAudio)
Bild
Bildschirmaufnahme (XSHM)
Browser
Diashow
Farbquelle
Fensteraufnahme (Xcomposite)
JACK-Eingabe-Client
Medienquelle
Sndio-Eingabe-Client
Szene
Text (FreeType 2)
VLC-Videoquelle
Videoaufnahmegerät (V4L2)
Gruppe



Neben der Liste an Quellen befindet sich ein Fenster mit den Audiogeräten, welche zu Beginn nicht angezeigt werden. OBS listet nur die Audio-Quellen auf, welche Live sind. Es gibt einen kleinen Trick, um die Anzeige der Audiogeräte im Mixer zu erzwingen. Zwischen dem Vorschaufenster und der Live Ansicht befindet sich ein kleiner Regler, welcher minimal nach rechts geschoben werden kann. Dieser Regler dient für den manuellen Übergang zwischen der Vorschau und dem, was gestreamt wird. Hierzu später mehr.





Im Audio-Mixer Fenster kann die Lautstärke des Audios angepasst werden, sollte zum Beispiel das Desktop-Audio das Mikrofon übertönen. Ein Audio Setup kann ziemlich komplex werden. Zum Beispiel möchte man sein Desktop Audio aufnehmen, jedoch nicht, was in einer Audiokonferenz mit mehreren teilnehmenden Personen besprochen wird. Hierzu werden virtuelle Soundkarten, Senken und des gleichen benötigt und ist von Betriebssystem zu Betriebssystem unterschiedlich zu konfigurieren.

Stream Einstellungen

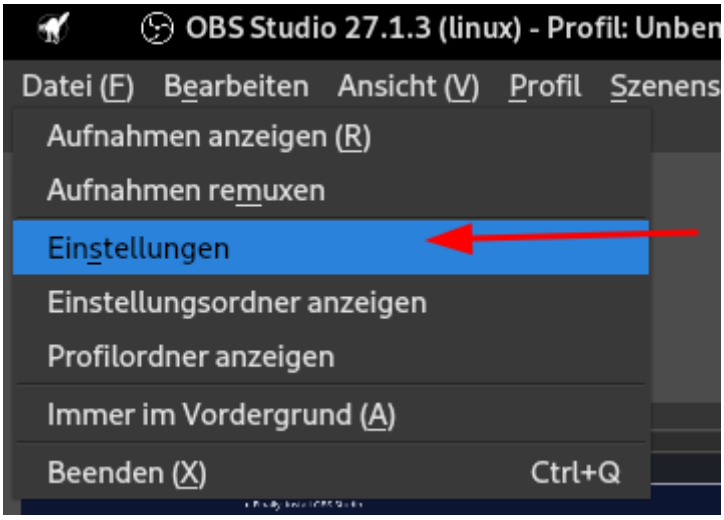
Da wir nun die **Bild-Quellen** und die **Audio-Quellen** definiert haben, muss nun das Ziel für den Stream definiert werden. Es gibt bereits voreingestellte Optionen für **Twitch**, **YouTube**, **Facebook**

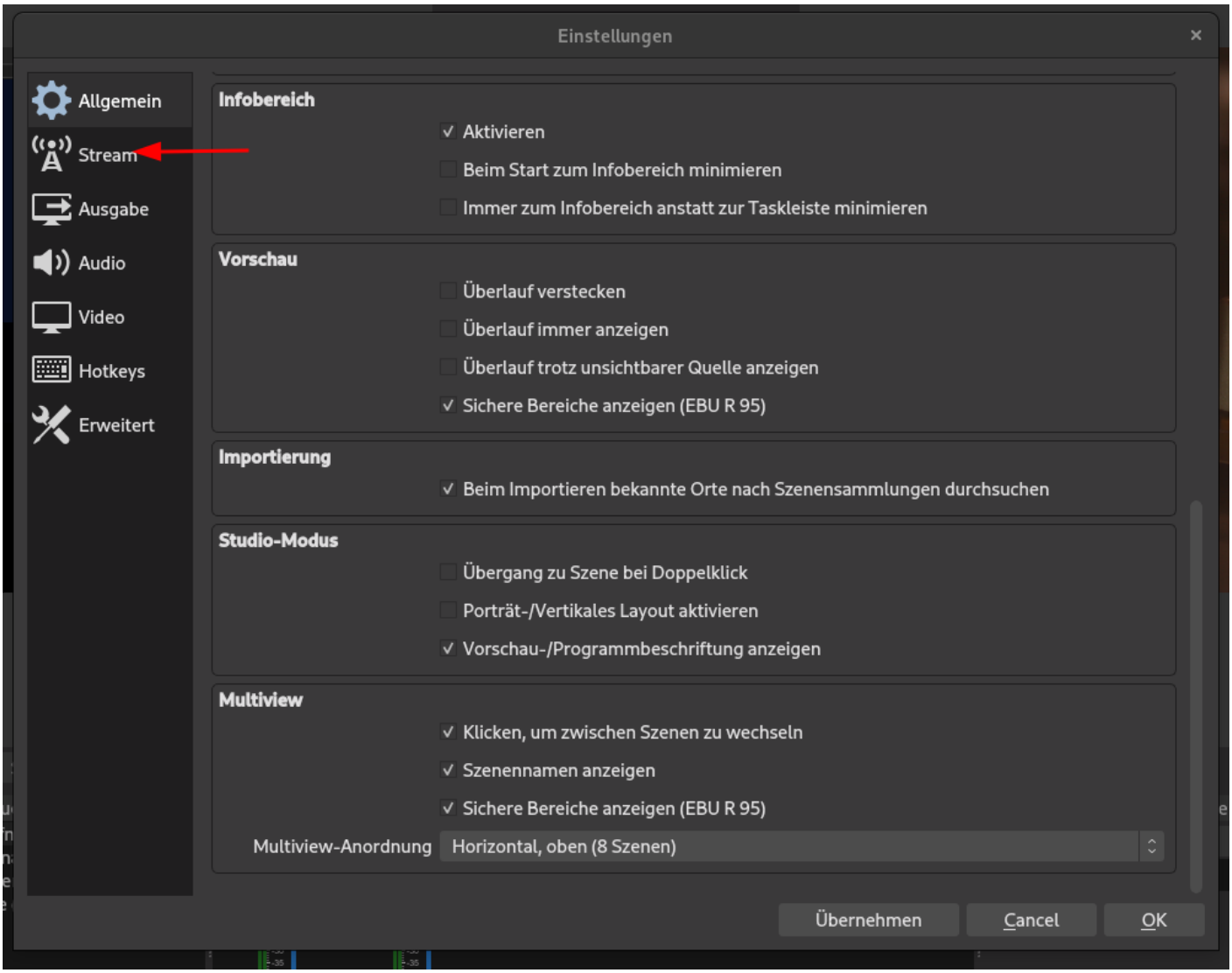
, **Twitter**, **Restream.io** und viele andere Anbieter. Es lassen sich aber auch nicht mit aufgelistete Dienste wie **media.ccc.de** einstellen. Die Konfiguration hierzu wird dann vom VOC mitgeteilt. Für dieses How To wählen wir Twitch.

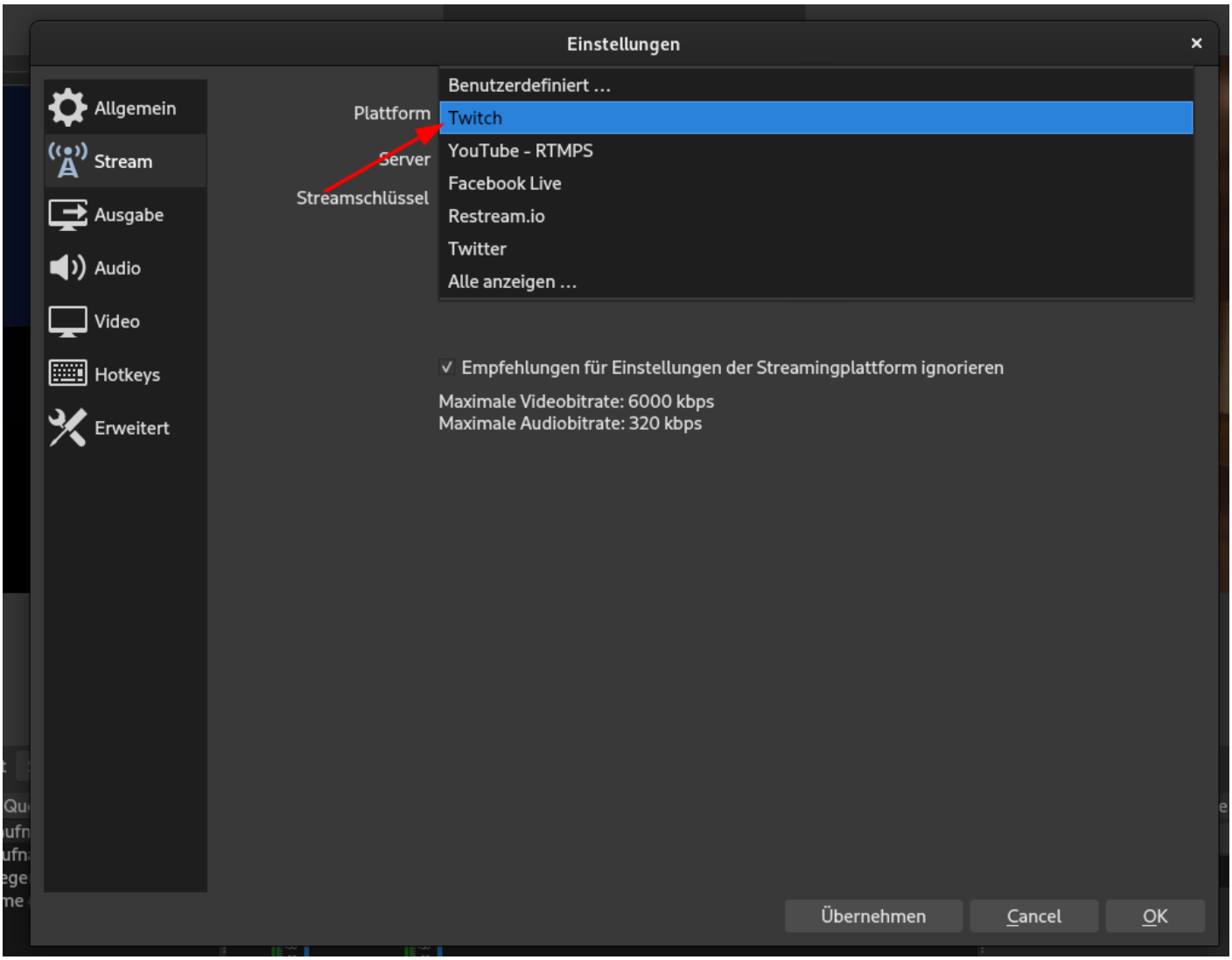
> Es wird ein Twitch Account benötigt

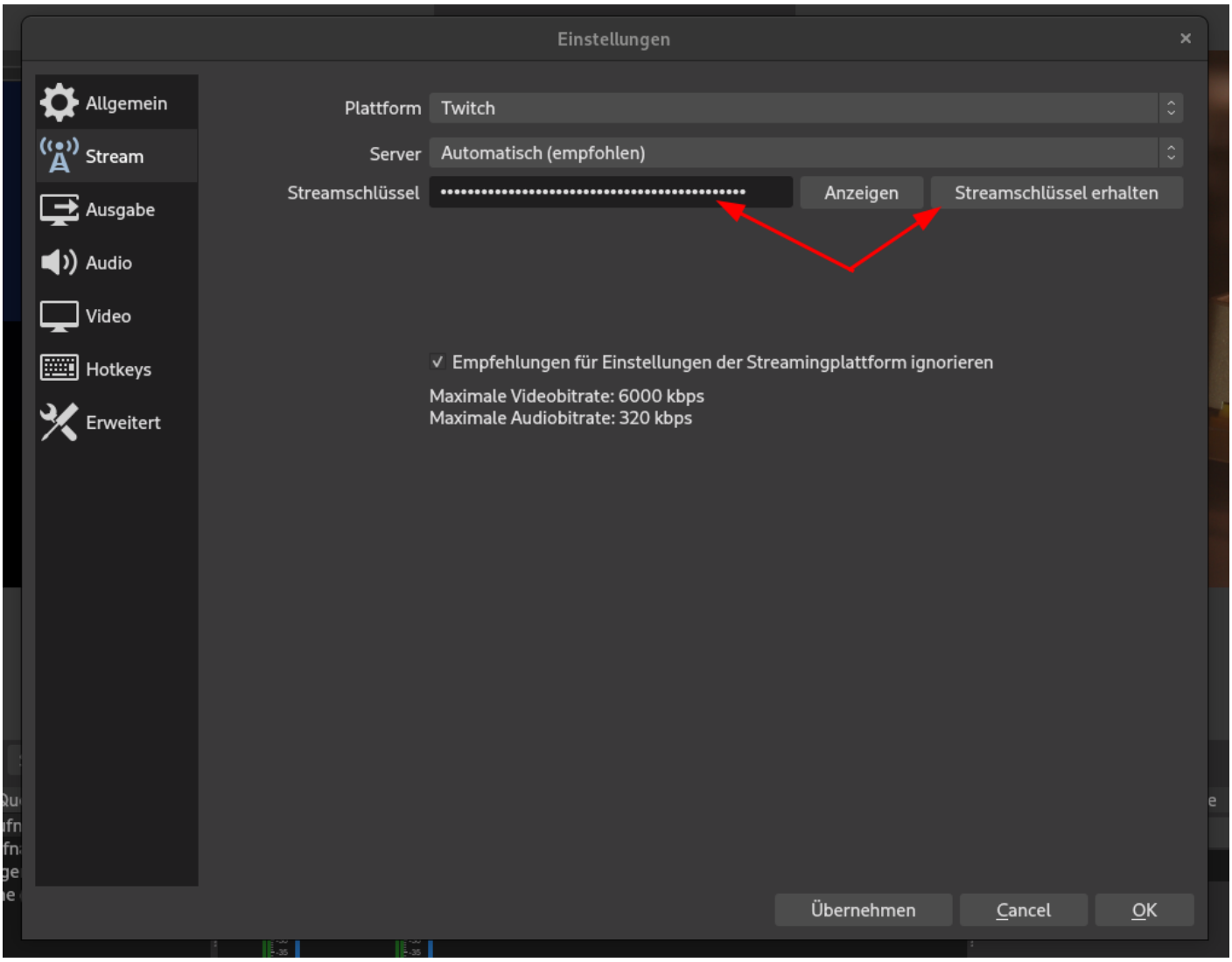
Die Anmeldung von OBS zu Twitch erfolgt nicht über die Zugangsdaten Usernamen/Email und Password, sondern über einen Schlüssel, welchen wir aus der Profilseite auf twitch.tv entnehmen können.

Twitch als Zielserver wird in den Einstellungen definiert. Geht hierbei über **Datei**/**Einstellungen** und klickt im neuen Fenster auf **Stream**. Über die Auswahl Plattform kann nun Twitch ausgewählt werden. Wenn ihr ein Browser offen habt und dabei in Twitch angemeldet seit, könnt ihr in OBS " **Streamschlüssel erhalten"** klicken und gelangt somit auf die korrekte Seite, um den Schlüssel zu kopieren, welcher in die Eingabemaske eingetragen werden muss. Damit weiß nun OBS wohin die Aufnahme gesendet werden soll, sobald man Live geht.




Einstellungen
Allgemein
Stream
Ausgabe
Audio
Video
Hotkeys
Erweitert
Infobereich
Aktivieren
Beim Start zum Infobereich minimieren
Immer zum Infobereich anstatt zur Taskleiste minimieren
Vorschau
Überlauf verstecken
Überlauf immer anzeigen
Überlauf trotz unsichtbarer Quelle anzeigen
Sichere Bereiche anzeigen (EBU R 95)
Importierung
Beim Importieren bekannte Orte nach Szenensammlungen durchsuchen
Studio-Modus
Übergang zu Szene bei Doppelklick
Porträt-/Vertikales Layout aktivieren
Vorschau-/Programmbeschriftung anzeigen
Multiview
Klicken, um zwischen Szenen zu wechseln
Szenennamen anzeigen
Sichere Bereiche anzeigen (EBU R 95)
Multiview-Anordnung
Horizontal, oben (8 Szenen)
Übernehmen
Cancel
OK


Einstellungen
Allgemein
Stream
Ausgabe
Audio
Video
Hotkeys
Erweitert
Plattform
Server
Streamschlüssel
Benutzerdefiniert …
Twitch
YouTube - RTMPS
Facebook Live
Restream.io
Twitter
Alle anzeigen …
Empfehlungen für Einstellungen der Streamingplattform ignorieren
Maximale Videobitrate: 6000 kbps
Maximale Audiobitrate: 320 kbps
Übernehmen
Cancel
OK



Video Einstellungen

Neben der Möglichkeit zu einem Anbieter zu streamen, besteht auch die Möglichkeit den Videostream lokal zu speichern. Dies ist ratsam, wenn man eine Sicherheitskopie des Rohmaterials haben will oder wenn ein Stream nicht gewünscht ist, sondern nur eine Aufnahme stattfinden soll. Dabei wird die Ausgabe, welche zum Streaminganbietern geschickt wird, lokal als Datei hinterlegt. Eine weitere Option ist, wenn der Anbieter wie Twitch eine maximale Videobitrate von 6000 kbps unterstützt, aber das lokal gesicherte Videomaterial höherwertige sein soll. Somit kann via Twitch ein Stream mit 6000 kbps gesendet werden, währen lokal das Videomaterial eine bessere Qualität hat. Die Einstellungen für den Stream und die Aufnahme wird in den Einstellungen unter Ausgabe definiert.

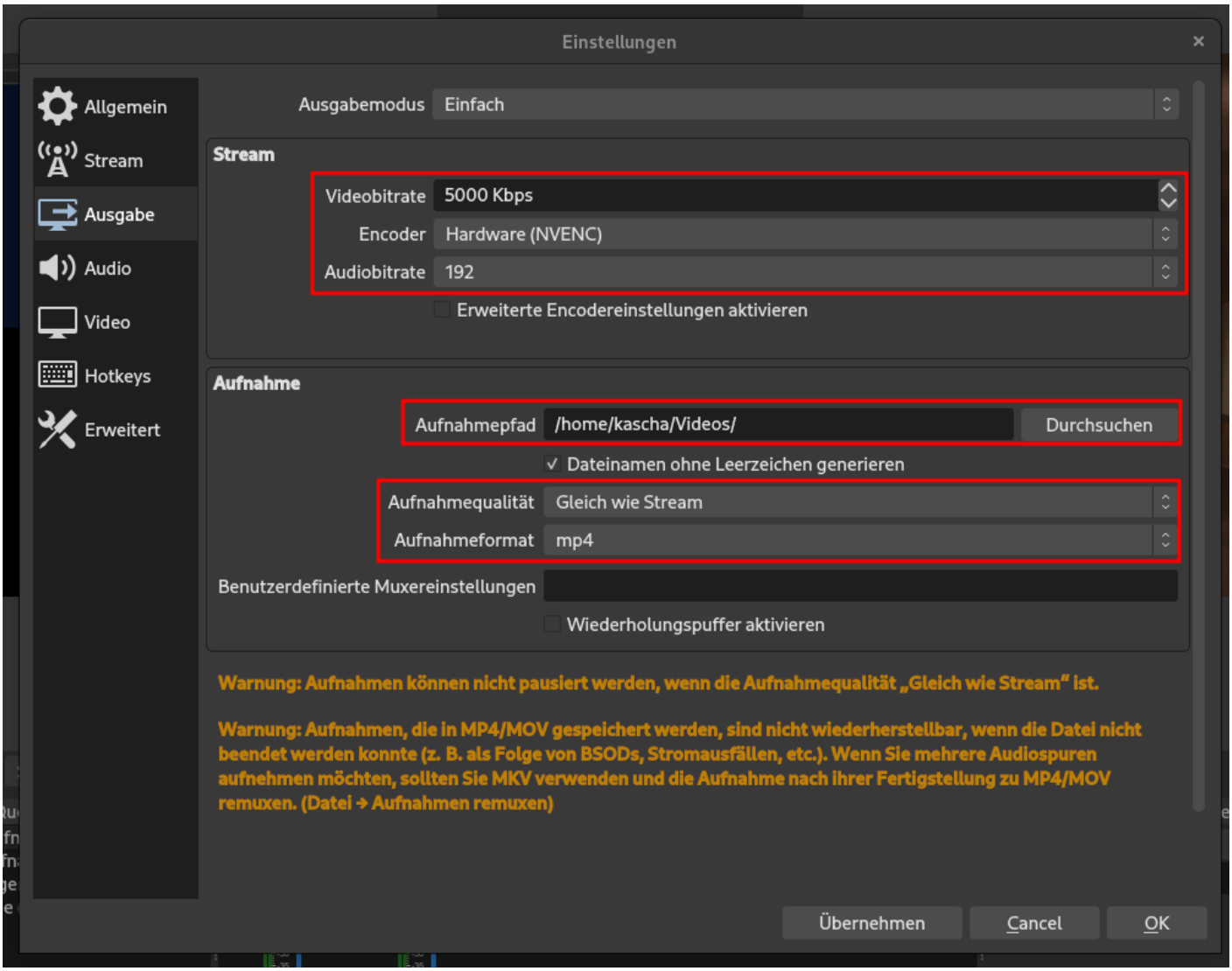
Stream

Unter dem Bereich Stream im Reiter Ausgaben, können verschiedene Einstellungen durchgeführt werden. Zum Beispiel welcher Encoder benutzt werden soll. Die Videobitrate für den Stream ist unteranderem von eurer Internetanbindung abhängig. Bei einer geringen upload Geschwindigkeit, muss das Video bevor es gesendet wird herunter gestampft werden. Dies bedeuetet, dass es zu Bildartefakten kommt oder der Stream matschig wird. Es ist Ratsam immer den Stream über eine

stabile Breitbandanbindung durchzuführen.

Für einen Stream mit einer Ausgabe von 1920x1080, einer Bitrate von 5000 Kbps, Audiobitrate von 192 und eventuell noch Gespräche via Mumble, Discord, Big Blue Button usw. sollten 7500 Kbps im Upload reichen. Als Faustregel für die maximale Bitrate kann man `Upload/1,5=Bitrate+Audiobitrate` nehmen.

Der Encoder welcher ausgewählt werden kann, ist abhängig von der Grafikkarte des Computers. Auf dem Screenshot kann man einen Hardware Encoder von Nvidia (NVENC) vernehmen. Dies kann bei euch abweichen, wenn ihr zum Beispiel eine Grafikkarte von AMD habt oder keine dedizierte Grafikkarte. Das Videoencoding kann auch Softwareseitig erfolgen, damit wird jedoch die CPU belastet.

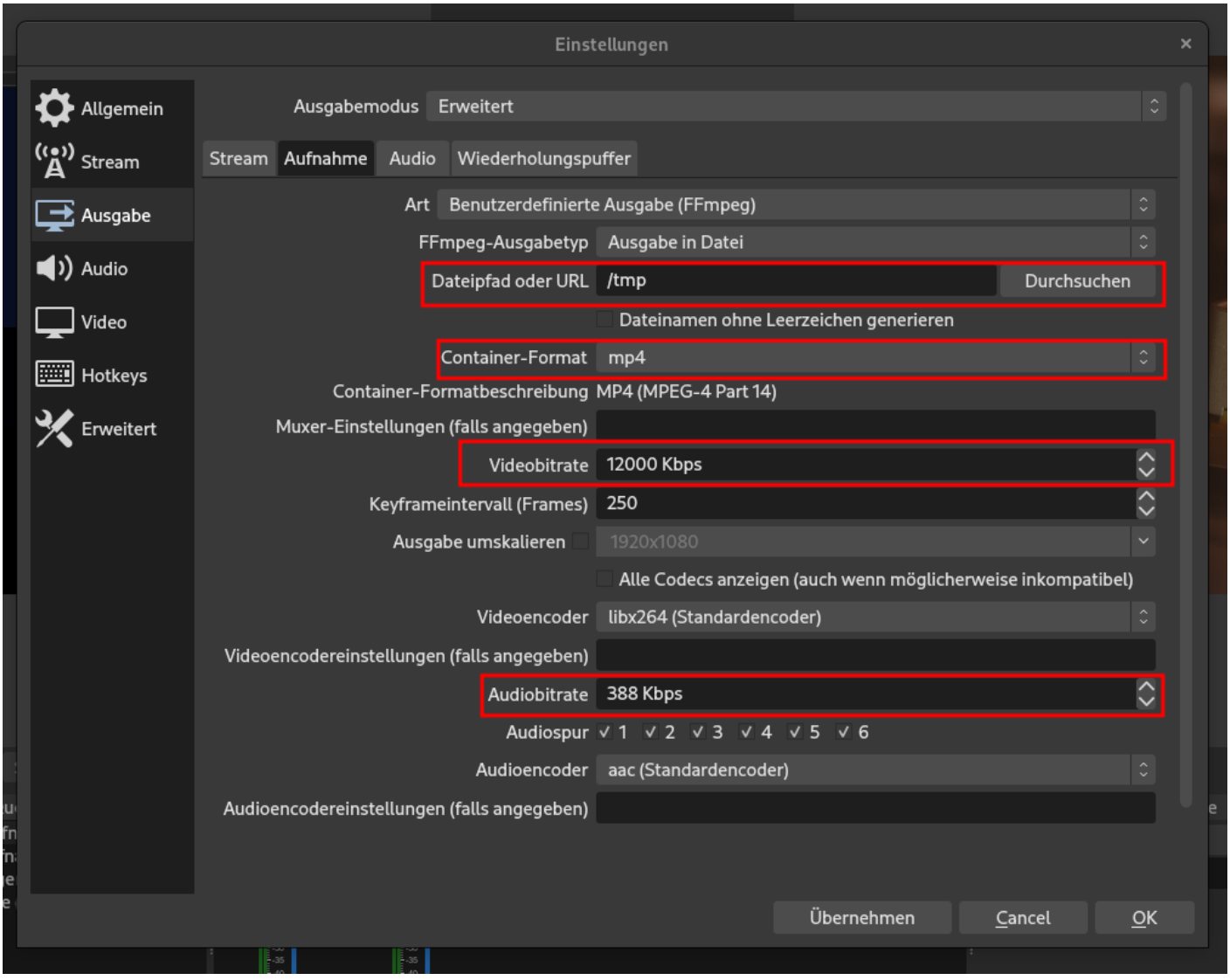


Aufnahme

Dem Screenshot oben, können die Aufnahme Einstellungen entnommen werden. Zu einem wohin die Videodateien gespeichert werden sollen, aber auch in welcher Qualität die Dateien gespeichert werden sollen. In der vereinfachten Ansicht, bietet OBS vorgefertigte Profile, welche von der Bezeichnung her selbsterklärend sind. Es besteht auch die Möglichkeit die Aufnahme frei zu konfigurieren, in dem man auf die Erweiterte Ansicht umstellt. Somit lassen sich auch der lokalen

Aufnahme verschiedene Tonspuren zuweisen. Ich rate für den Anfang davon ab, die Erweiterte Ansicht zu benutzen, da man bei OBS mit den einfachen Einstellungen auf der sichereren Seite ist.

Wie dem Screenshot oben bereits entnommen werden kann, gibt es eine Warnung bei der Nutzung von MP4 als Videoformat. Die Technische erklärung für die Warnung ist sehr simpel. Eine MP4 Datei kann im Gegensatz zu anderen Videoformaten keine Fehler beinhalten (zum Beispiel, fehlen pLötzlich einige Sekunden in der Mitte des Videos). Sobald das encodieren des Videos nicht ordentlich beendet wurde, ist die gesamte Datei unbrauchbar. Ein MKV Container hingegen kann in mitte der Aufnahme Fehler haben oder kurzweilig die Aufnahme stoppen. Der Nachteil einer MKV ist, dass dieses Format nicht überall unterstützt wird. Der einfachheit halber und der besseren Unterstüzung hat sich MP4 durchgesetzt. Somit ist es ratsam MKV-Datei aufzunehmen und nach der Aufnahme in MP4 umzuwandeln.

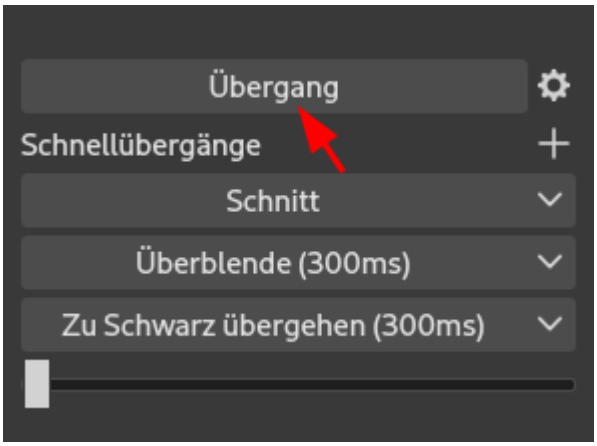


Live gehen und Aufnahme

Da wir alle Bild und Audio-Quellen hinzugefügt haben, welche wir streamen möchten und OBS die Video und Stream Einstellungen konfiguriert haben, können wir mit dem Streamen beginnen. Wählt die Szene aus, welche ihr Streamen möchtet und klickt zwischen dem Vorschau und Programm

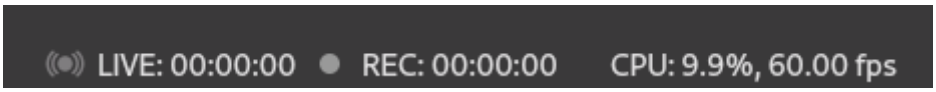
Fenster auf Übergang um aus der Vorschau das Livebild zu machen und klickt dann unten Rechts auf einer oder mehrere der Start Funktionen.

- Stream starten
  - Startet den Stream zum eingestellten Streaminganbieter (in diesem Beispiel Twitch).
- Aufnahme starten
  - Die Szene wird lokal gespeichert.
- Virtuelle Kamera starten
  - Die Szene wird als Virtuelle Kamera ausgegeben und kann dann zum Beispiel im Big Blue Button als Webcam benutzt werden.





Sofern keine Fehlermeldung kommt, ist der Stream Live. Um zu sehen ob der Stream oder die Aufnahme (noch) läuft, zeigt OBS die dauer der Aufnahme und Streams im Footer an. Sollte der Stream unterbrochen sein, wird dies in dieser Anzeige dargestellt. Als weitere Sicherheit, kann die Seite, auf der der Stream läuft aufgerufen und gecheckt werden (benötigt weitere Bandbreitenkapazität). Zur Beendigung von der Aufname oder dem Stream klickt man auf die Schlatfläche, wo zuvor noch starten stand: Aufname oder Stream stoppen.



Revision #52
Created 2022-03-14 16:39:18 UTC by Kascha
Updated 2022-05-10 09:15:50 UTC by ronja