

Streaming des Desktop-Inhalts auf einen anderen PC

Version: 1.0, Autor: „Guenther“, im Februar 2022

Vorbemerkung

Aufgabenstellung:

Es soll von einem PC/Notebook im heimischen LAN der Desktop mittels Datenstrom auf einen andern PC im selben LAN gespiegelt werden. Dabei sollen sowohl Bildschirminhalte als auch Tonausgaben, aber auch das Bild einer evtl. vorhandenen Kamera (z.B. beim Laptop) und des eingebauten Mikrofons übertragen werden.

Der andere PC empfängt den Datenstrom und gibt ihn auf dem Bildschirm wieder.

Lösungsansätze:

Grundsätzlich erfordert die Aufgabenstellung die Einrichtung eines Streamingdienstes bzw. -servers auf dem PC sowie eine Möglichkeit, den Desktop inkl. Tonausgabe zu „capturen“ (wie man so schön auf Neudeutsch sagt).

Ansatz 1: VLC (Video LAN)

Aufgrund eines Hinweises eines Anwenders habe ich zunächst die Software VLC verwendet. Diese kann in der aktuellen Version 3.0.16 (Februar 2022) bereits von Haus aus Daten streamen. Ich habe damit erfolgreich Videodateien und auch Netzwerkstreams mittels des HTTP-Protokolls direkt an einen zweiten PC gestreamt.

Was jedoch leider nur teilweise gelang, war die Übertragung des Desktops. Hier kann VLC nur den Bildschirm übertragen, der Ton bleibt weg. Es gibt zwar ein zusätzliches Tool (Github Screen Capturer), mit dem man Bildschirm und Ton streamen kann. Dies hat jedoch bei mir ebenfalls nicht korrekt funktioniert, insbesondere wurde kein Ton übertragen.

Es mußte also ein anderer Ansatz her.

Ansatz 2: Open Broadcaster Software Studio (OBS)

Da VLC die Aufgabe nicht zufriedenstellend erfüllt, liegt es nahe, eine Alternative zu verwenden. Dazu gibt es reichlich Software, jedoch nur wenige kostenlose Projekte, die auch den Desktop streamen können. Ich habe mich hier für die "Open Broadcaster Software Studio" software"(OBS) entschieden und diese installiert.

OBS kann alles streamen, was man sich nur wünschen kann, auch parallel. OBS erwartet jedoch als Ziel seines Streamings einen Streaming-Server. Für diese Aufgabe wählte ich NGINX.

Als Streamingprotokoll habe ich das RTMP (*real time messaging protocol*) ausgesucht.

Da ich zur Lösung der og. Aufgabe keine allzugroßen Ansprüche an den Streaming-Server hatte, wählte ich den NGINX-Server. Bis zur Version 1.7.12.1 ist in der Windows-Version ein RTMP-Modul bereits eingebaut und man muß nichts sonst installieren. Daher habe ich bequemerweise diese Version installiert.

Mit dieser Kombination (OBS + NGINX) ist es sehr einfach, den Desktop, die Kamera, das Mikrofon und was man sonst noch möchte, zu streamen. Die Qualität des Streams hängt stark vom jeweiligen Equipment des PCs oder Notebooks ab und ist bei mir zwar verbesserungswürdig. Jedoch habe ich es im ersten Versuch gleich mit 60 fps hinbekommen und konnte ein Youtube-Video auf dem 2. PC abspielen, das zeitgleich auf meinem Laptop lief.

Im folgenden nun die einzelnen Installationsschritte unter Windows. Ich verwende Win 10.

Kurzfassung der Installation

Installationsschritte:

1. Herunterladen und Installieren von Open Broadcaster Software Studio (OBS)
2. Herunterladen und Installieren von NGINX V 1.7.12.1
3. Konfigurieren des NGINX-Servers als RTMP-Server
4. Konfigurieren von OBS zum Streamen an den NGINX-Server
5. Test

Hinweis: Diese Anleitung ersetzt nicht bestimmte Grundkenntnisse im Umgang mit PCs, Streaming generell sowie der Installation und Einrichtung von Software unter Windows. Es wird vorausgesetzt, daß diese Kenntnisse vorhanden sind. Ich übernehme keinerlei Verantwortung für die Korrektheit dieser Anleitung bzw. das korrekte Funktionieren der Soft- oder Hardware, nachdem die Installationsschritte durchgeführt wurden. Ich kann jedoch versichern, daß die nachfolgende Installation auf meinen PCs mit Windows 10 funktioniert.

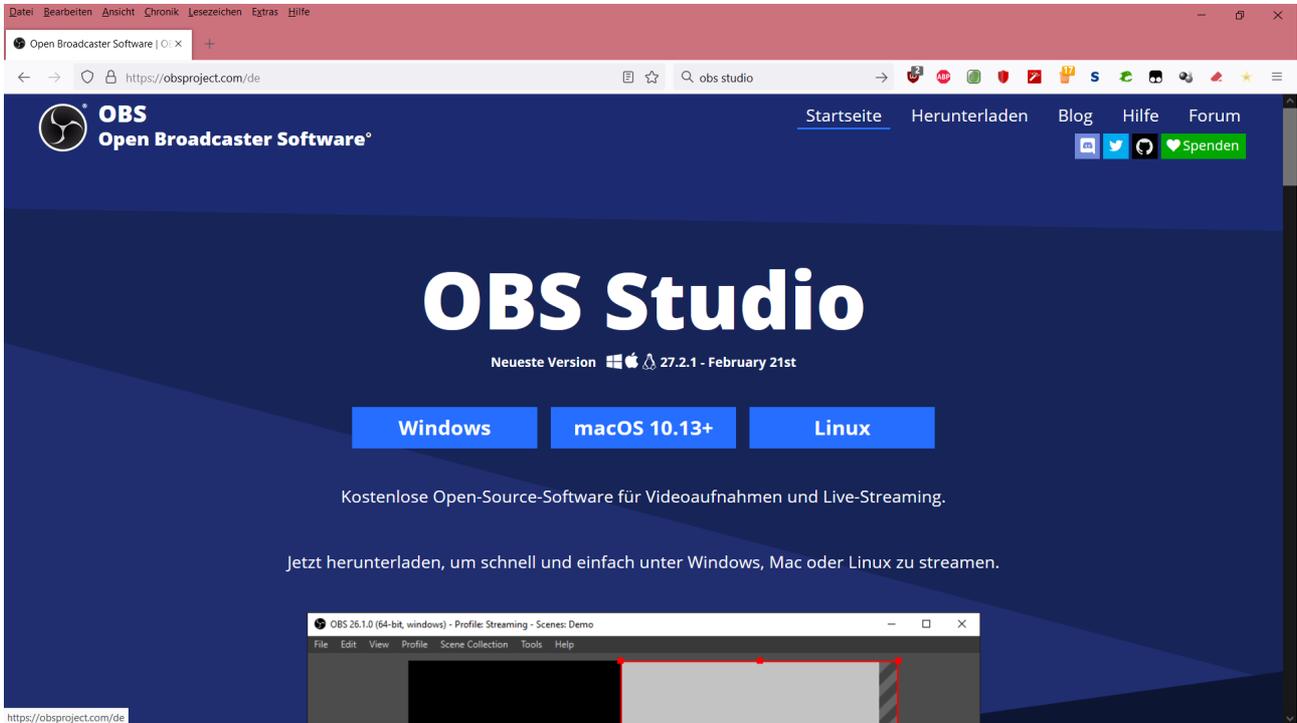
Ferner rate ich dringend dazu, sich mit der Software OBS eingehender zu beschäftigen, um alle Möglichkeiten effizient nutzen zu können. Dazu gibt es auch Video-Tutorials.

Und nun: Auf geht's!

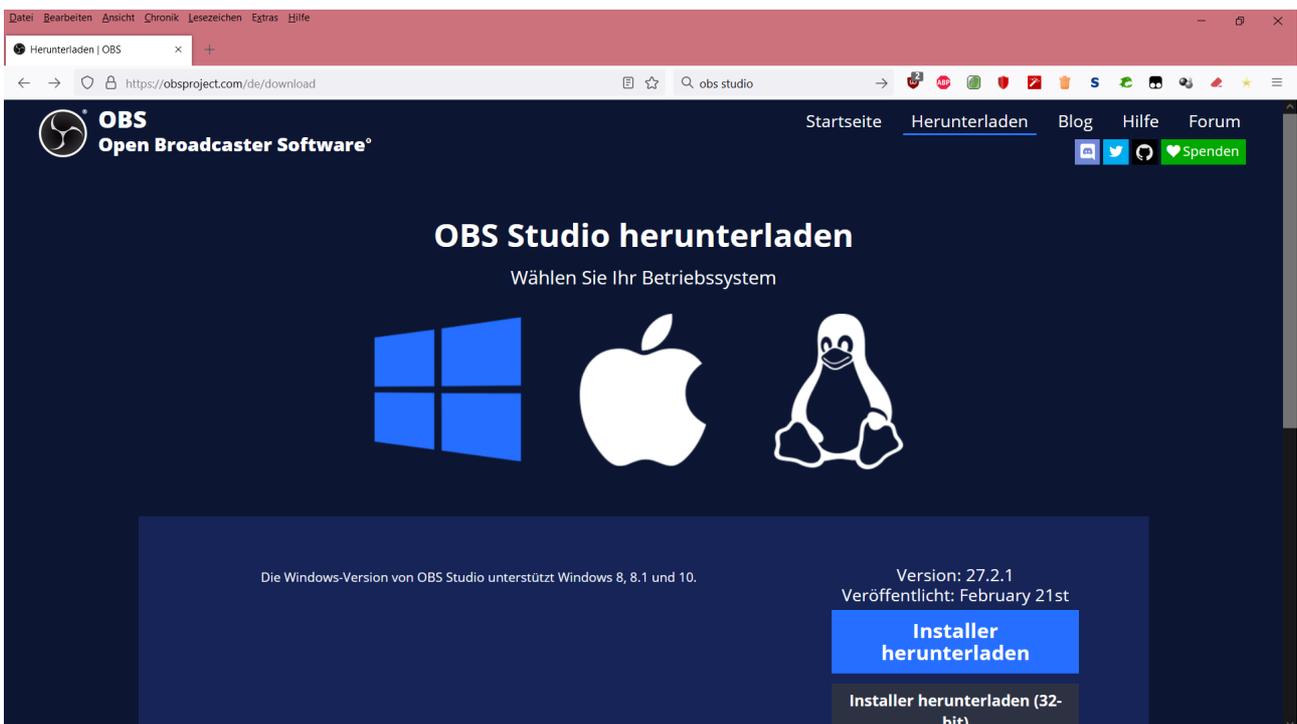
Installationsschritt 1: Herunterladen und Installieren von Open Broadcaster Software Studio (OBS)

Webseite:

<https://obsproject.com/>



Dort auf „Download“ gehen, dann die passende Version (Windows) herunterladen:



Ich habe die *Installer-Version* gewählt: *OBS-Studio-27.2.1-Full-Installer-x64.exe*

Diese Datei ist ca. 112MB groß und direkt ausführbar.

Nach dem Download rate ich zu einem Virenscan der Datei mit einem aktuellen Scanner.

Installation:

Die og. Installerdatei als Administrator ausführen. Einfach die Voreinstellungen übernehmen und als Abschluß dann OBS 27.2.1. ausführen lassen.

Bei der Ersteinrichtung werden gleich eine Menge Fragen gestellt, die man jetzt oder auch später beantworten kann. Ich habe die Einrichtung später vorgenommen und die Ersteinrichtung an dieser Stelle abgebrochen, da ich zunächst den RTMP-Server *NGINX* installieren wollte.

Wir machen also weiter mit dem:

Installationsschritt 2: Herunterladen und Installieren von NGINX v1.7.12.1

Webseite zum Download:

https://link-4share.com/download/1_TheQvTce/nginx_1_7_12_1_lizard_zip.html

Die ZIP-Datei heißt: *nginx 1.7.12.1 Lizard.zip* und ist 2691 KB groß.

Nach dem Download rate ich zu einem Virenscan der Datei mit einem aktuellen Scanner.

Installation:

Das ZIP-Archiv in ein beliebiges Verzeichnis entpacken. Ich habe meins *nginx* genannt und direkt im Windows-Hauptverzeichnis erstellt: *C:\nginx*.

Installationsschritt 3: Konfigurieren des NGINX-Servers als RTMP-Server

Nach dem Herunterladen und Entpacken des ZIP-Archivs muß man NGINX dafür konfigurieren, als RTMP-Server zu werkeln. Dazu sind zwei Schritte erforderlich:

A) Die Konfigurationsdatei *nginx.conf* muß im Unterverzeichnis *C:\nginx\conf* erstellt werden. Dazu startet man den Explorer (den Verzeichnis-Manager, nicht den Internet Explorer) und wechselt in das Unterverzeichnis *conf* und kopiert die dortige Datei *nginx-org.conf*. Die Kopie nennt man *nginx.conf*.

B) Man muß nun diese *nginx.conf* editieren (notepad.exe genügt). Folgende Zeilen müssen eingefügt werden:

```
rtmp {
    server {
        listen 1935;
        chunk_size 4096;
        application live {
            live on;
            record off;
        }
    }
}
```

Diese Zeilen werden eingefügt nach der Stelle:

```
events {
    worker_connections 1024;
}
```

Diese Stelle ist ganz oben in der *nginx.conf*.

Der Dateianfang mit den eingefügten Zeilen sieht dann so aus:

```
#user nobody;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid logs/nginx.pid;

events {
    worker_connections 1024;
}

rtmp {
    server {
        listen 1935;
        chunk_size 4096;

        application live {
            live on;
            record off;
        }
    }
}

http {
    include mime.types;
    default_type application/octet-stream;

    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    # '$status $body_bytes_sent "$http_referer" '
    # '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log logs/access.log main;

    sendfile on;
    #tcp_nopush on;

    #keepalive_timeout 0;
    keepalive_timeout 65;

    #gzip on;
```

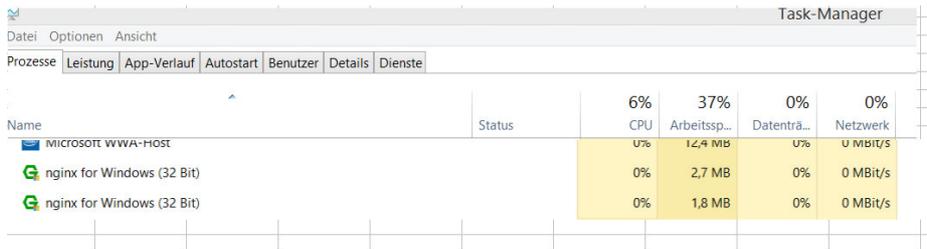
usw....

Die Datei nun speichern. Das war's schon.

Start des NGINX-Servers:

Ausführen der Datei `nginx.exe` im Verzeichnis `C:\nginx` (oder wo Ihr das ZIP-Archiv halt entpackt habt).

Da passiert scheinbar nichts, es geht vielleicht kurz ein Fenster auf (`cmd.exe`), das sich gleich wieder schließt. Der Server wurde dennoch gestartet. Man kann ihn im *Task Manager* sehen:



Name	Status	6% CPU	37% Arbeitssp...	0% Datenträ...	0% Netzwerk
Microsoft WWW-Host		0%	12,4 MB	0%	0 MBit/s
nginx for Windows (32 Bit)		0%	2,7 MB	0%	0 MBit/s
nginx for Windows (32 Bit)		0%	1,8 MB	0%	0 MBit/s

Damit ist die Einrichtung des NGINX-Servers abgeschlossen.

Installationsschritt 4: Konfigurieren von OBS zum Streamen an den NGINX-Server

Als nächstes müssen wir OBS so einrichten, daß es den Bildschirm „captured“ und als Stream per RTMP-Protokoll an den NGINX-Server schickt. Dieser sorgt dann dafür, daß alle Geräte im LAN (Heimnetzwerk) diesen Stream empfangen können.

Damit dient der NGINX quasi als Streaming-Plattform, ganz so wie z.B. Youtube. Das Prinzip ist so:

- OBS erzeugt den Screen Capture und konvertiert diesen in einen Stream.
- OBS sendet den Stream kontinuierlich an den NGINX-Server.
- Der NGINX-Server stellt den Stream „zur Abholung bereit“.
- Der zweite PC kontaktiert den NGINX-Server per RTMP und holt sich dort den Stream.

Richten wir nun also OBS entsprechend ein.

Hinweis: Die nachfolgenden Screens weisen die OBS-Version 26.1.1 aus, was daraus resultiert, daß ich diese Anleitung bereits letzten Januar erstellt habe. Die Bedienung und die Einstellungen haben sich jedoch in der Version 27 nicht geändert und können so übernommen werden, wie ich sie dargestellt habe.