Thank you for choosing PLAYDECK! We are convinced that with PLAYDECK you will significantly increase your productivity, whether you produce live events, stadium TV, theatre performances, museums, TV broadcasts or virtual and hybrid streaming events. You can run PLAYDECK both on-site and cloud-based.

PLAYDECK provides you with 8 completely independent output channels in which you can organise your videos, graphics, audio files, live inputs, streams or YouTube clips simply by dragging and dropping.

PLAYDECK supports almost any codec and converts all content to the selected output format in real time. The output is completely flexible via dedicated output cards (e.g. from Blackmagic, AJA, Bluefish, …), via the extended desktop, via NDI or via Streams.

In the Studio Edition, your recording channel can be edited while the recording is still running (edit-while-ingest, e.g. highlight editing), played back with a time delay or streamed. For maximum flexibility you can choose from a wide range of different recording formats.

But PLAYDECK (depending on your licence) is much, much more than a simple playout system:
You can trim and crop your content, you can automatically adjust the volume, you can schedule the start of your contributions, you can control the playlist externally, you can start playback via your video switcher, you can transfer closed captions, you can completely customise PLAYDECK with advanced scripting functions, you can…

You are welcome to inform yourself about the incredible functionality of PLAYDECK in the following chapters. We are sure you will love it!

Your PLAYDECK PC Build

PLAYDECK can be installed on any Windows 64-bit machine. This includes Windows Server Versions or Custom Cloud Builds.

PLAYDECK needs fast modern Hardware, if pushed to the Limit. It has to transcode any given Video Format and Framerate in Real-time into your selected Output Format. Ideally without any Frame Drops.

As with all Tasks, it all depends on your use case: Are you using the LITE Edition to operate one Full HD Channel in a Live Event for some Hours? Or are you using STUDIO to broadcast multiple UHD Channel to several receiver eg Streams, NDI, and all in a 24/7 manner?

One thing is certain: Without a powerful modern NVIDIA GPU you will likely be disappointed. Dont try to run PLAYDECK over a Intel Onboard GPU, it is not designed for that. Your NVIDIA GPU should have ideally at least a Score of 8000 on the Passmark Scale.

All other PC components should not be much older than 2 years, just to meet modern driver standards, as we update PLAYDECK at least 4 times a year with the latest driver. You will want to utilize and profit from that, as all PLAYDECK updates are free of charge, as long as you have a valid license.

Here a the Specs we use for our own PLAYDECK production builds:

- BeQuiet Straight Power 11 750W
- Gigabyte Z790 AORUS Elite AX So.1700 Dual Channel DDR5 ATX Retail
- Intel Core i7 13700F 16 (8+8) 2.10GHz So.1700 TRAY
- Noctua NH-U9S Tower Cooler
- 32GB Corsair Vengeance black DDR5-5200
- 500GB Samsung 970 EVO Plus M.2
- 1TB Samsung 970 EVO Plus M.2
- 12GB Gigabyte Geforce RTX 4070 Windforce OC Active PCIe 4.0 x16 (Retail)
- Blackmagic Design DeckLink SDI 4K

Some additional informations about Hardware in relation to PLAYDECK:

Output Card

IF you use a dedicated Output Card (eg BM Decklink), you will not only offload ressources to the Card and reduce overall System GPU/CPU load, but you will also benefit from:

- More “true” Colors
- Nearly Zero Frame Drops due to Frame Rate Control
- Overall higher Picture Quality
- Much more stable than HDMI over Desktop (no Windows-interference)

We support Output Cards from these Manufacturers (see complete List):

- Blackmagic Design
- AJA
- Deltacast
- Bluefish444
- DekTect
- Magewell
- Osprey
- Stream Labs
- Yuan
- ASIO Devices (eg DANTE)

CPU Codecs, Keying and Alpha Channel

Please note, that these Video Codecs can not be decoded via GPU currently. PLAYDECK will fall back to CPU for these Codecs, so plan a strong CPU, if you use these Codecs regulary:

- ProRes
- HAP
- DNxHD

If you are using Keying or Alpha Channel in genereal, and you do not need the extreme high quality of ProRes, try to use the HAP-A Codec instead. It will still be decodec on the CPU, but use alot less CPU power.

Notebooks

If you are aiming for mobile productions, we go with the Razor Blade Notebook and extend it with a Blackmagic Ultra Studio card via the thunderbolt interface. If in doubt, which manufactorer to go for, decide for a Gamer Notebook. Those are designed to run at maximum performance. Avoid buying Office-type Notebooks like Dell, which are designed for power saving and can seriously limit your playout performance.

Supported Input/Output Devices

We basically support all cards of the following manufacturers, since we always include the latest drivers. But because of the vast amount of cards out there, we only tested the most common ones.

Blackmagic Design

Important: Needs Version 14.5+ of Blackmagic Desktop Video Setup.

- DeckLink 4K Extreme 12G
- DeckLink 8K Pro
- DeckLink Duo 2
- Intensity Pro 4K
- DeckLink Mini Monitor / Mini Recorder
- DeckLink Quad 1 / 2 / HDMI Recorder
- DeckLink SDI 4K
- DeckLink Studio 2 / 4K
- Ultra Studio HD Mini / 4K Mini / 4K Extreme 3 / Monitor 3G
- DeckLink IP/SDI HD

AJA

- Corvid 24 R1
- Corvid 44
- Corvid 88
- KONA LHi
- KONA IP

Deltacast

- DELTA-3G-e 22
- DELTA-3G-elp-d 8c
- DELTA-3G-elp-key 11

Bluefish444

- Epoch 4K Neutron
- Epoch 4K Supernova S+

- Epoch Neutron
- Epoch Supernova CG
- KRONOS K8

DekTec

- DTA-2144B

Magewell

- Pro Capture Quad HDMI
- Pro Capture Quad SDI

Osprey

- Osprey 915
- Osprey 925
- Osprey 927
- Osprey 935
- Osprey 945
- Osprey 914
- Osprey 924
- Osprey 944
- Osprey 1214
- Osprey 1215
- Osprey 1225
- Osprey 1227
- Osprey 1245
- Osprey 1285
- Osprey M15
- Osprey M14
- Osprey M24
- Osprey M25

Stream Labs

- Alpha HD
- MH4LM
- MS4
- MSP2

Yuan

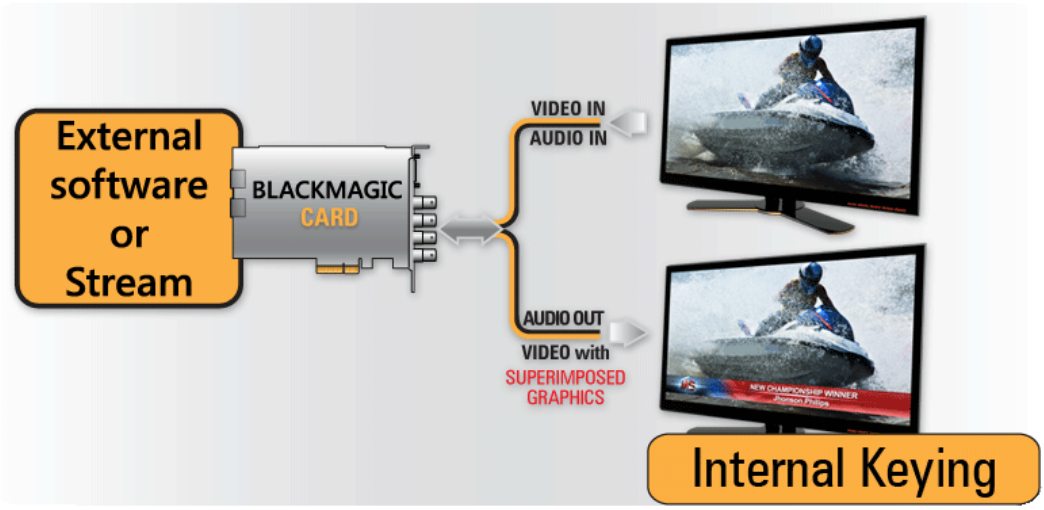
- SC550N1

Internal and External Keying

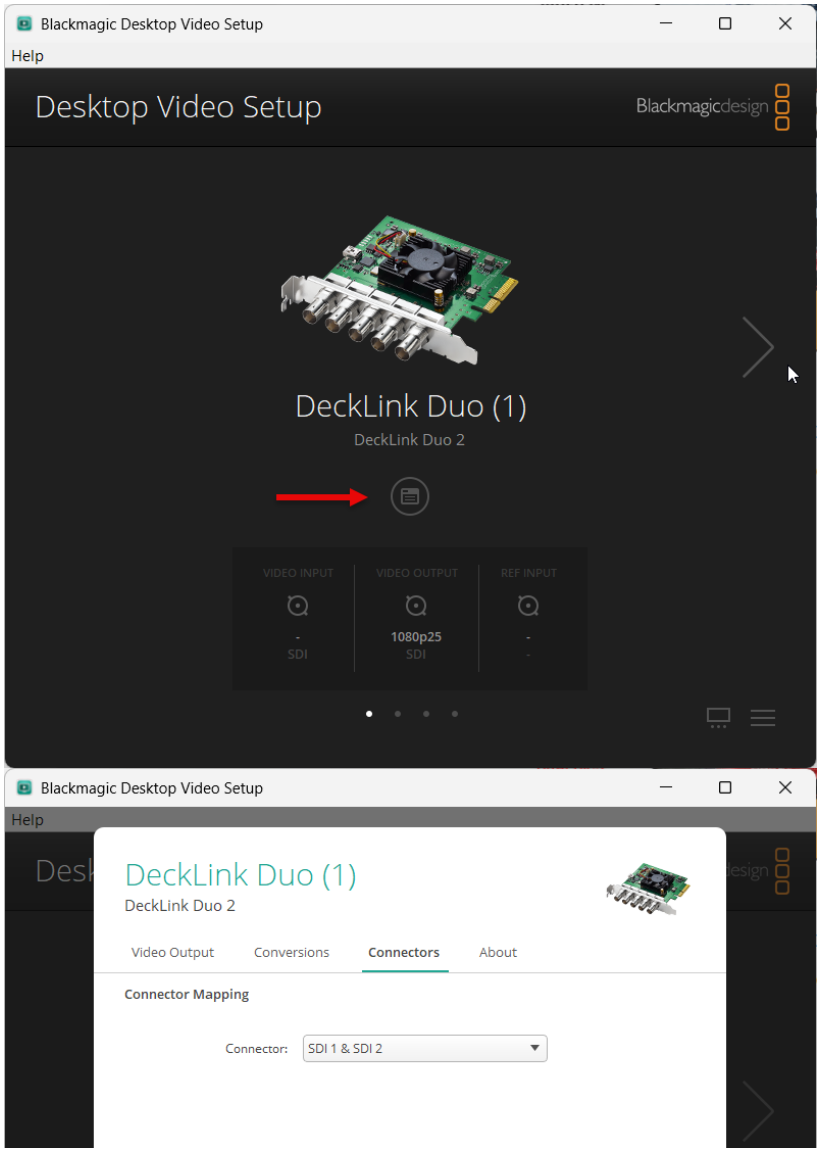
PLAYDECK supports both Internal and External Keying, given that the Output card supports it too. You can also output the key or fill signal seperatly.

Internal Keying

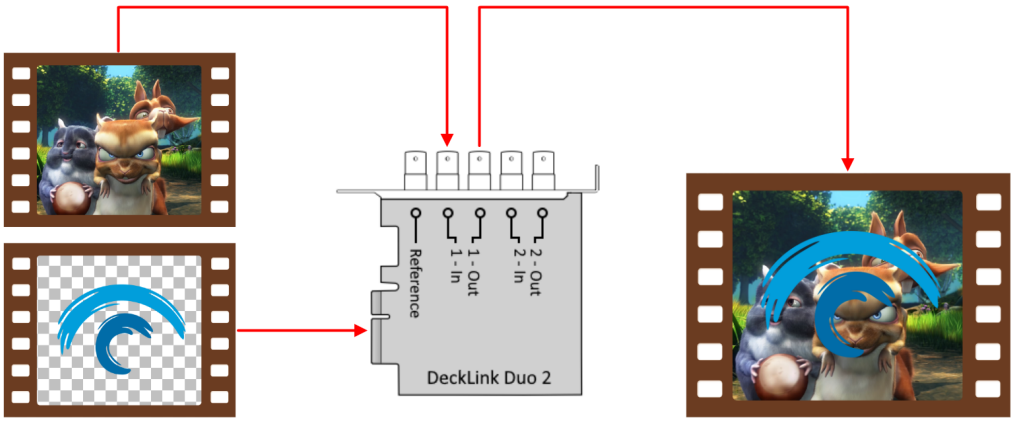
In Internal Keying mode, PLAYDECK will superimpose the video over an incoming signal on the same card:



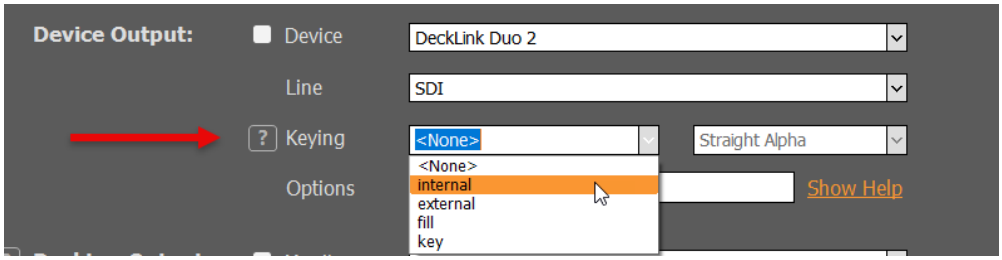
Lets take the Decklink Duo 2 for example: It has 4 SDI Ports. We need to tell the Card, which 2 Ports will be used for Internal Keying with the help of Desktop Video Setup (Blackmagic’s own Setup-App):



The Decklink will now operate like this:



If we start PLAYDECK now, the INTERNAL Option becomes selectable as Device Output:

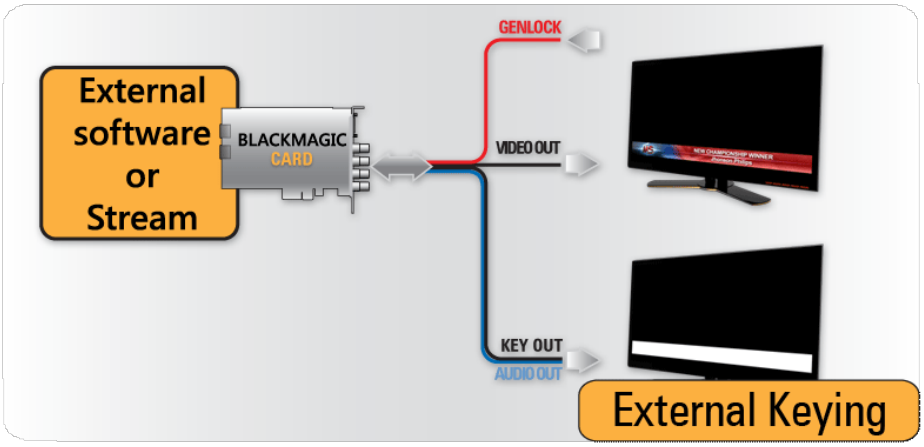


Once you activate the Device for Keying, PLAYDECK will automatically change your Background to TRANSPARENT and your Color Space to ARGB32 (to support Alpha Channel):

The Image shows PLAYDECK Overlays Sample: Create Overlay > HTML > Screen-Title.html.

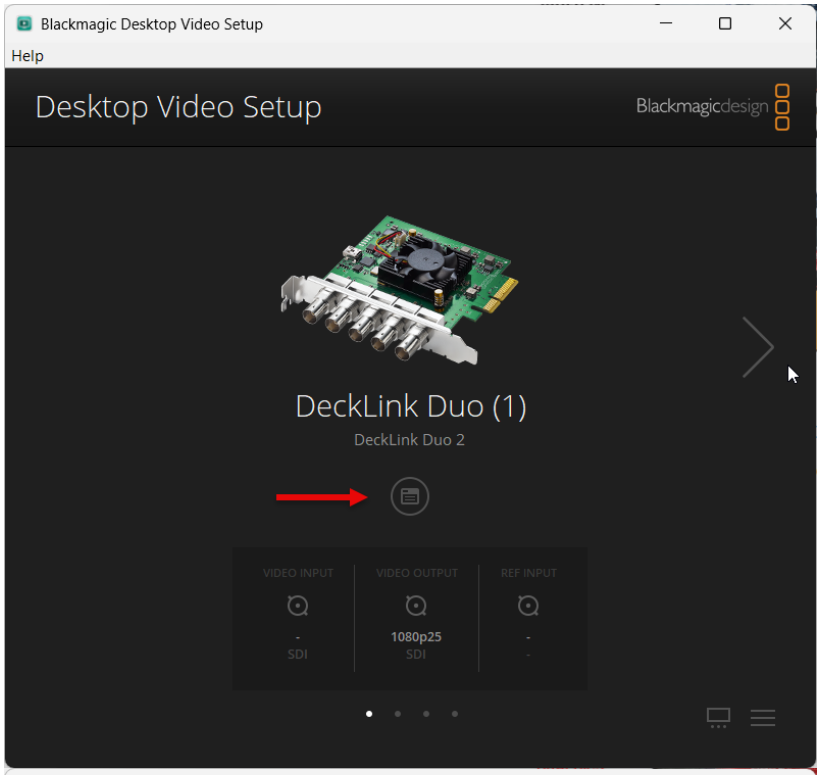
External Keying

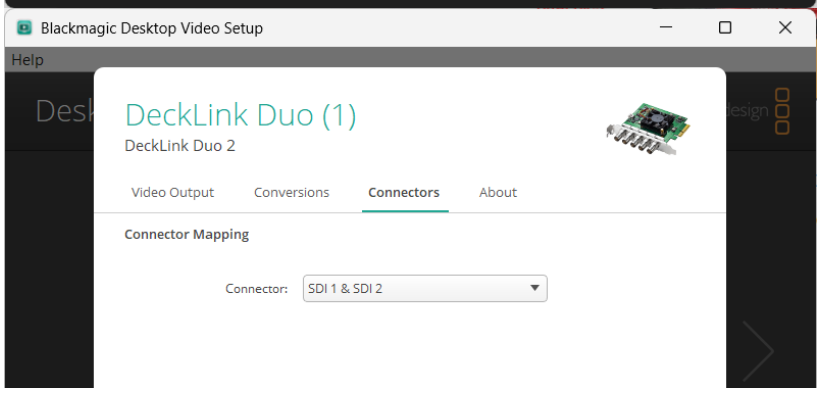
In External Keying mode, PLAYDECK generates both Fill and Key signals and the keying is made by an external keyer:



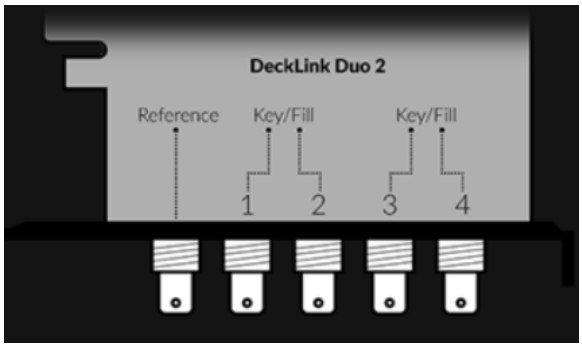
Lets take the Decklink Duo 2 for example: It has 4 SDI Ports. We need to tell the Card, which 2 Ports will be used for External Keying with the help of

Desktop Video Setup (Blackmagic’s own Setup-App):

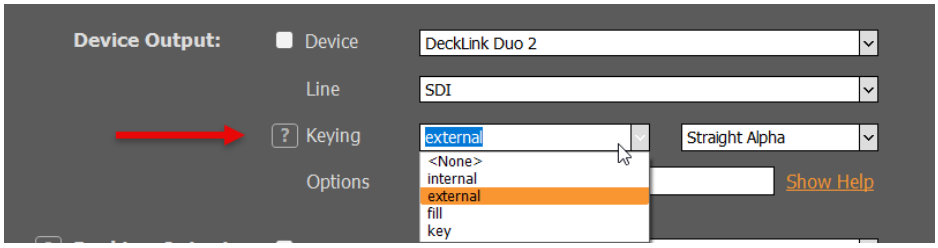




The Decklink will now operate like this:



If we start PLAYDECK now, the EXTERNAL Option becomes selectable as Device Output:



Once you activate the Device for Keying, PLAYDECK will automatically change your Background to TRANSPARENT and your Color Space to ARGB32 (to support Alpha Channel):

The Image shows PLAYDECK Overlays Sample: Create Overlay > HTML > Screen-Title.html.

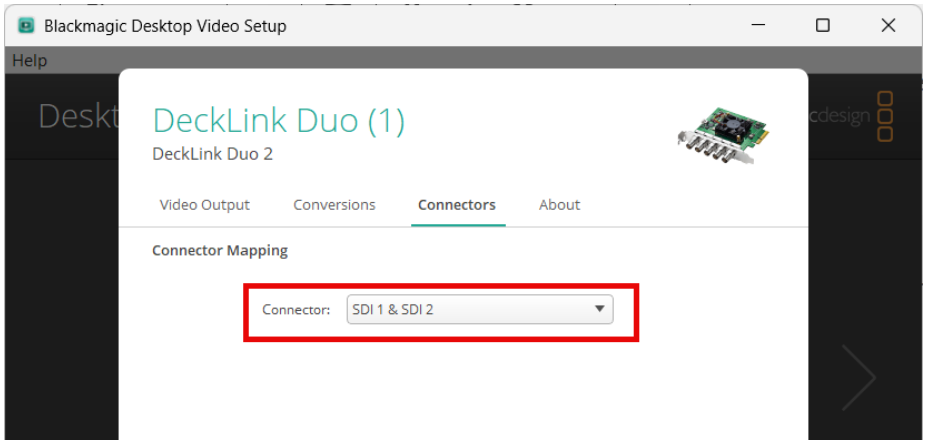
PiP with Internal Keying

This article will show how to utilize Internal Keying as means of overlaying Video over Video (Picture in Picture).

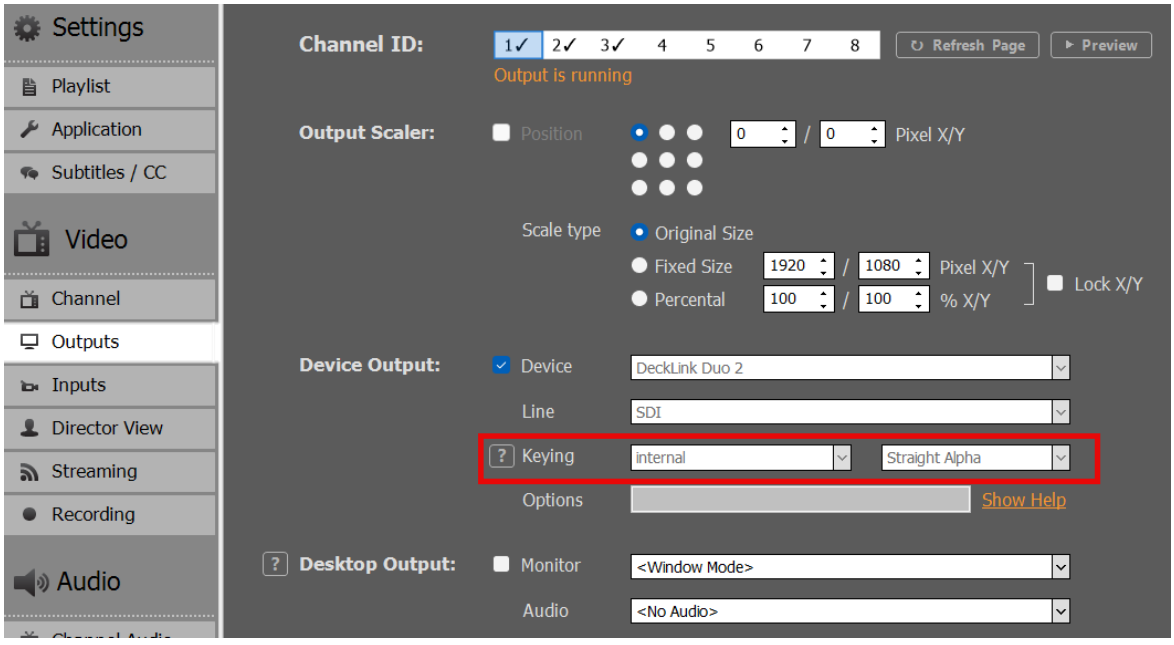
Setup of Internal Keying

We need to setup PLAYDECK to have a BG FEED and a PIP FEED and mix them into the MAIN FEED. For this we use Internal Keying. See this article for more info on Internel Keying.

Use any Device that is capable of Internal Keying. In our example we use the Blackmagic DeckLink Duo2. Open the Dektop Video Setup and combine and first 2 SDI Outputs. Leave the other Outputs solo:



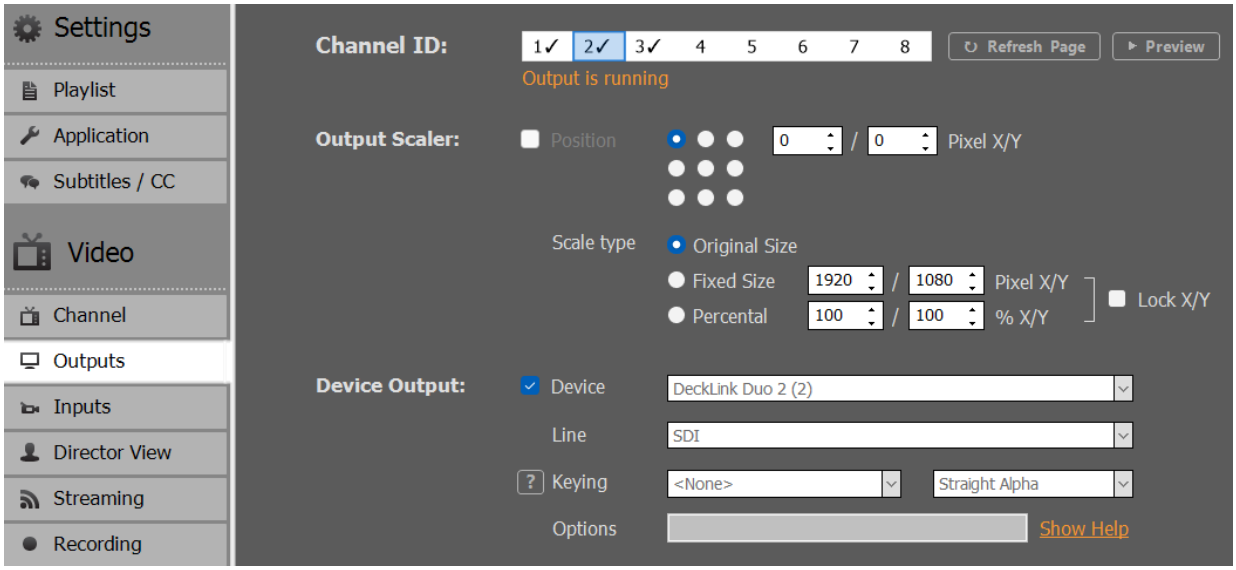
Next we enable Internal Keying in Channel 1 in PLAYDECK:



Setup of BG FEED

Now we need to send our BG FEED to SDI 1. For this we use Channel 2 and send it via SDI Loop from SDI 3 to SDI 1. Please note, that because we combined

SDI 1+2, SDI 3 is now designated as “Decklink Duo 2 (2)”:



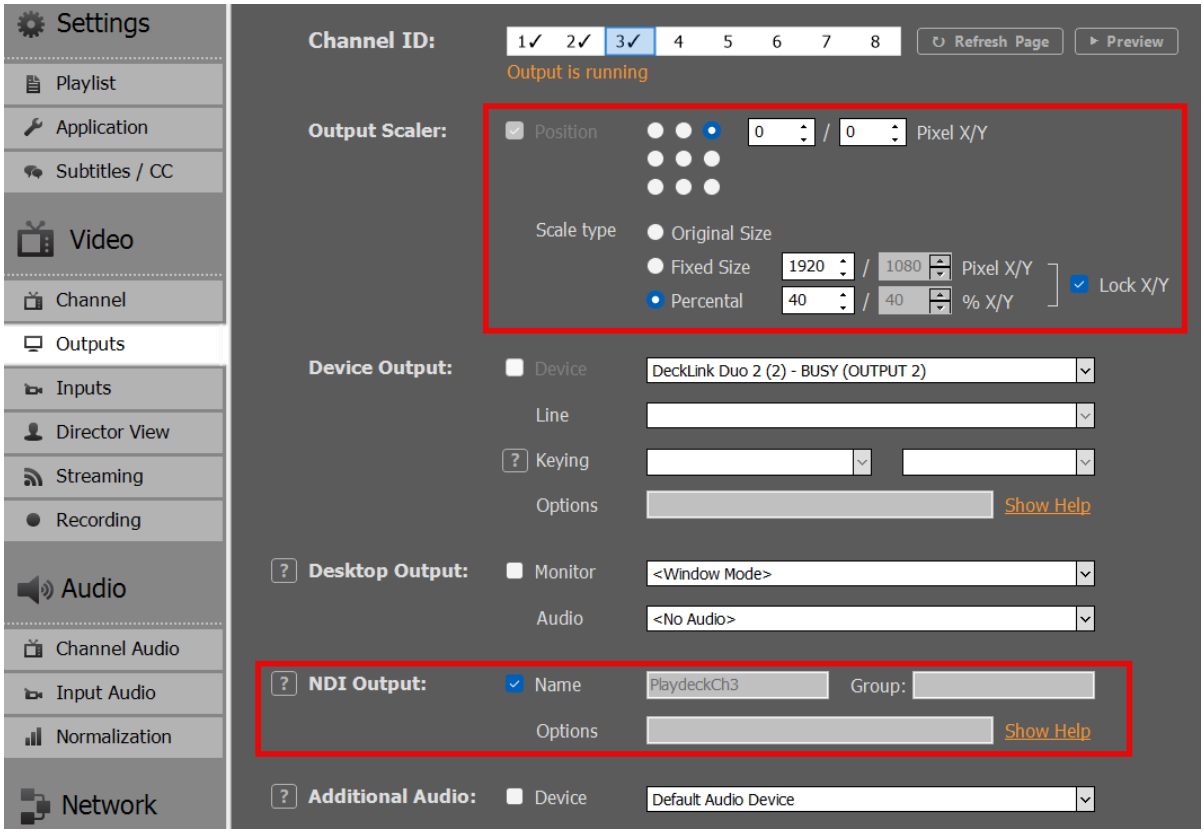
Then we start any video content on Channel 2:

If you have a SDI Monitor on SDI 2, you can already see the BG FEED playing there now.

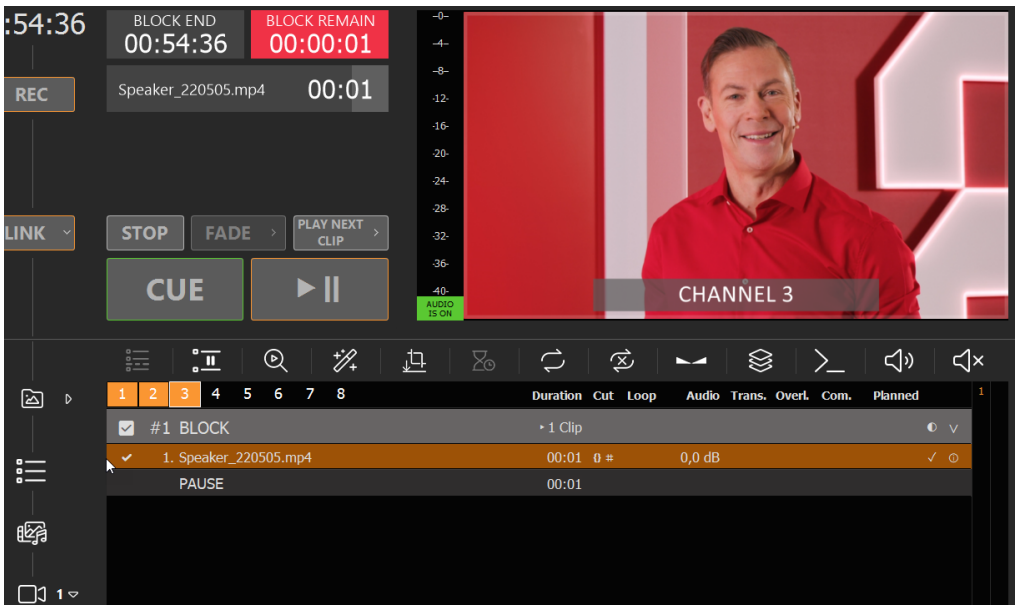
We could now go ahead and play any Content with Alpha on Channel 1, which is keyed over the BG FEED, e.g. Alpha Videos (ProRes, HAP-A) or simple Overlays. But since we need our PIP FEEDBACK scaled and positioned, we need to set it up on another Channel.

Setup of PIP FEED

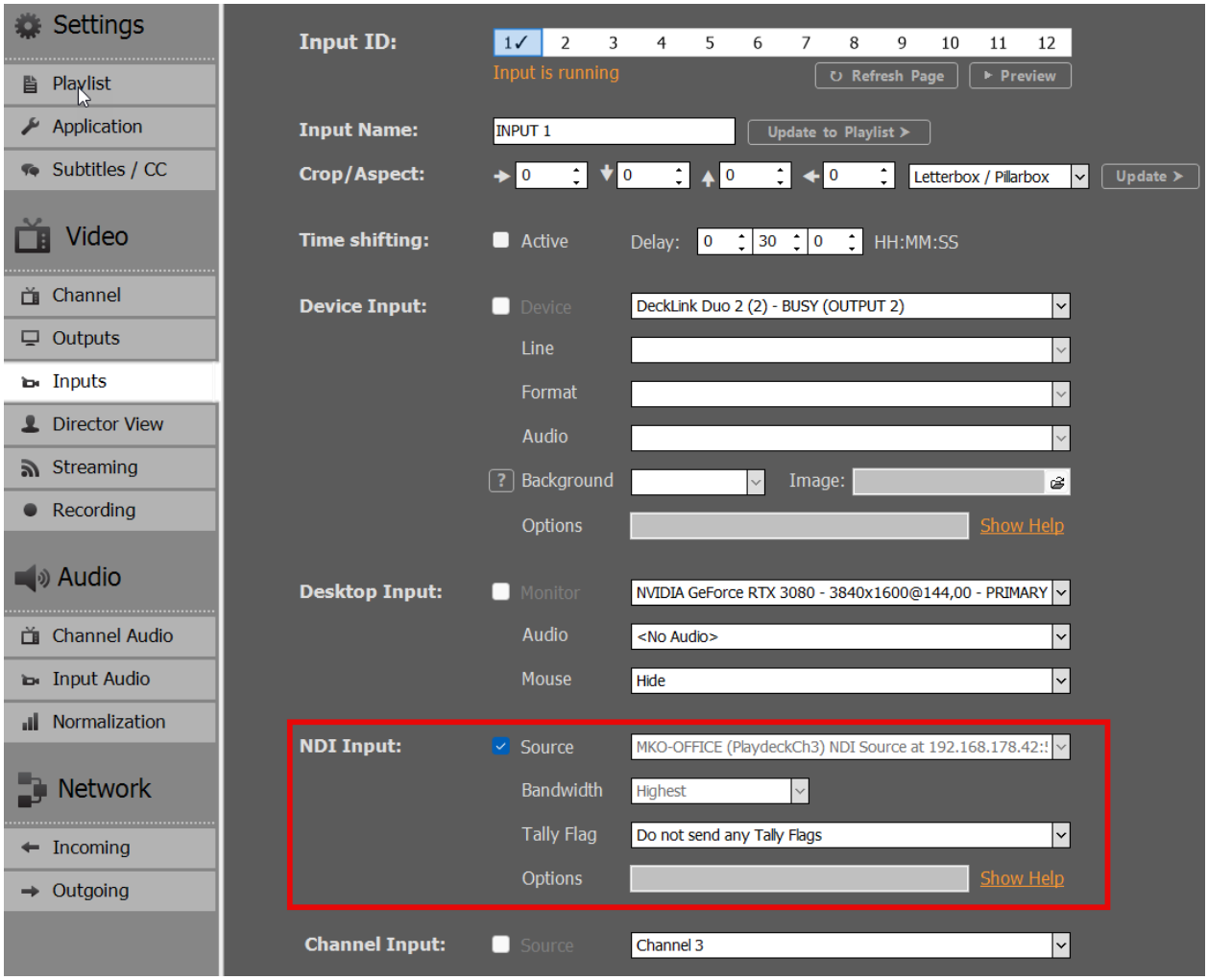
We use Channel 3 as PIP FEEDBACK, so we can play regular Video Content here. We want to send it via NDI from Channel 3 to Channel 1 and activate the Output Scaler, so that our PIP is in the UPPER RIGHT Corner:



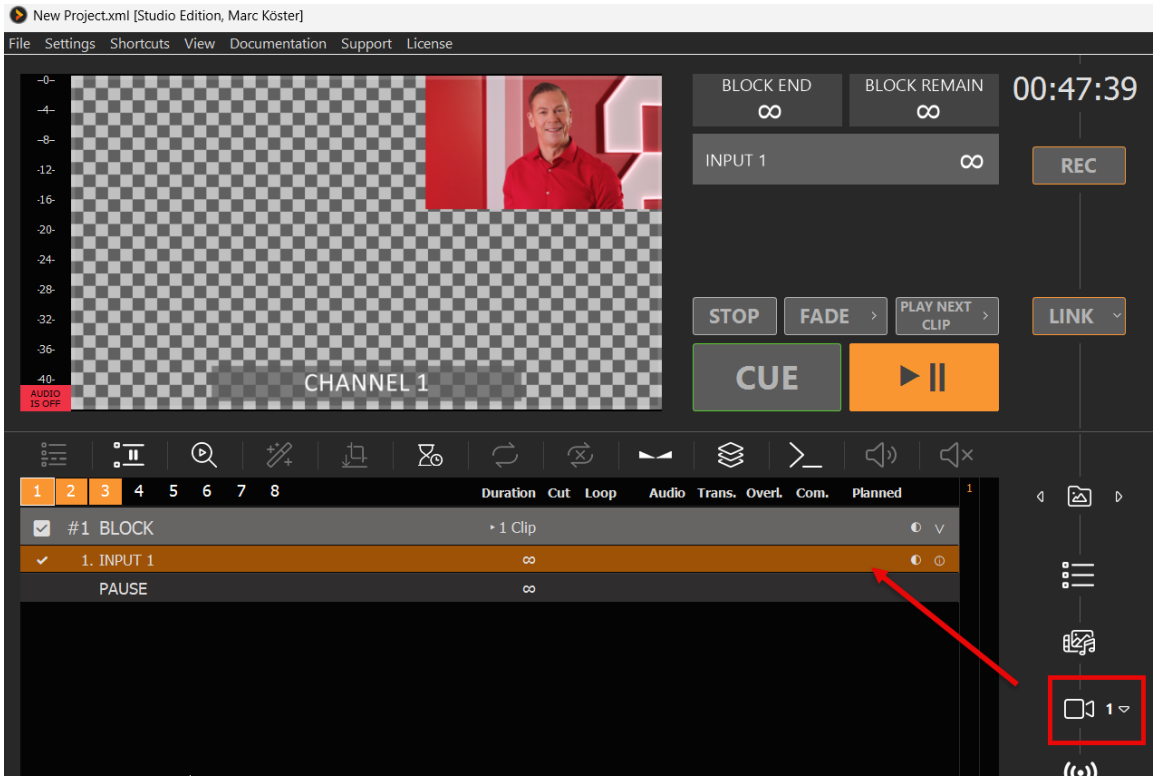
We now insert a Video Clip on Channel 3, but this could also be Live Video, Streams or anything else:



We loop this NDI Feedback to Input 1:



Then insert Input 1 into the Playlist of Channel 1:



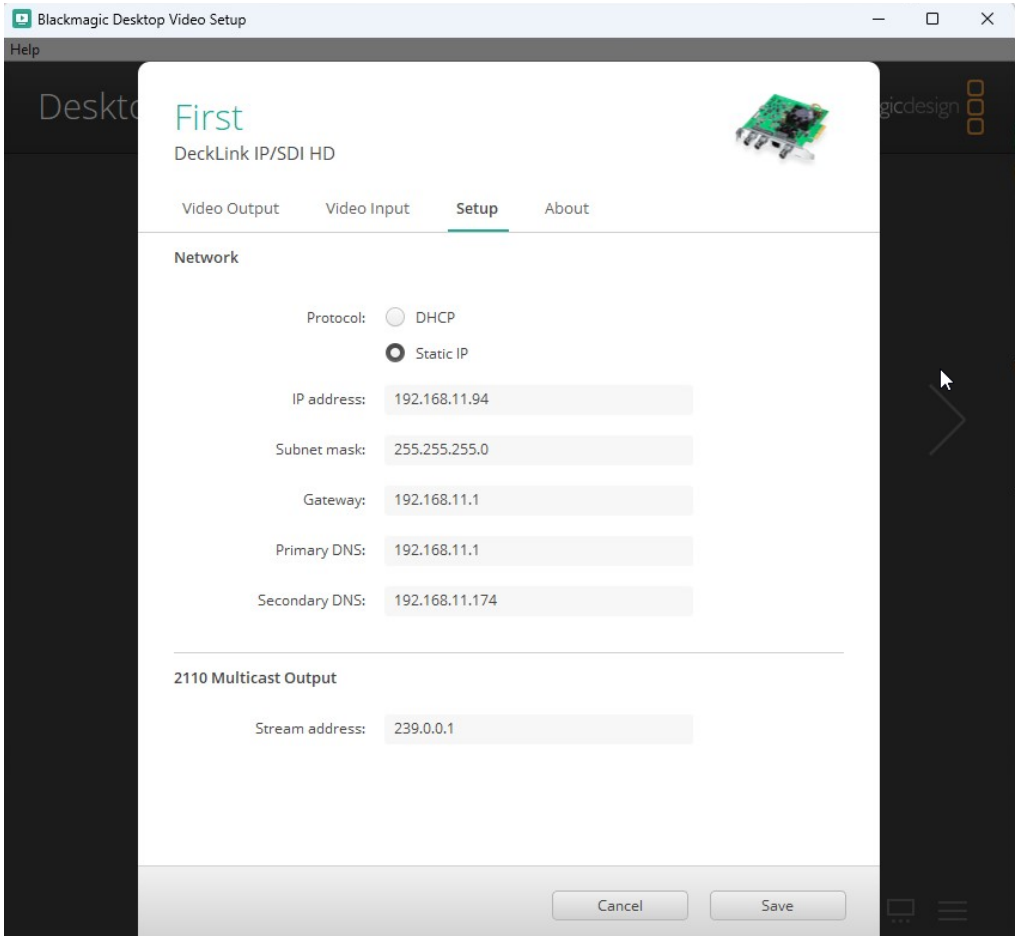
We now downscaled the Channel 3 Playlist into Channel 1, which will be keyed over the BG FEED of Channel 2.

FINAL OUTPUT / MAIN FEED

The MAIN FEED is send to SDI 2 and this is the result on the SDI Monitor of SDI 2 (photographed):

ST 2110 Device Setup

Most 2110 Devices can be configured with the Tools given by their manufacturer. For example the “Blackmagic DeckLink IP/SDI HD” can be setup with the Blackmagic Desktop Video Setup:



For high quality and lag-free network operation, we recommend using high end network hardware (router, switch) as well as high speed ethernet cables (cat6+).

The specific network configuration will not be covered in this article, as there is too much diversity.

AJA 2110 Cards

PLAYDECK allows to assign a network configuration file, which will be loaded upon PLAYDECK start.

Structure of Configuration File (JSON)

```
{
    "protocol":"2110",
    "network2110":
```

```
    { // this part configures the parameters of the device itself, the
connectors (SFP) with IP configurations
        "ptpPreferredGMID":"00-00-00-00-00-00-00-00",
        "ptpDomain":0,
        "setup4k":"false",
        "multiSDP":"false",
        "audioCombine":"false",
        "rxMatchOverride":0,
        "sfps":
        [   // the configuration itself for each of the connectors
            {
                "designator":"sfp1",
                "ipAddress":"192.16.45.8",
                "subnetMask":"255.255.0.0",
                "gateWay":"255.255.255.255",
                "enable":"true"
            },
            {
                "designator":"sfp2",
                "ipAddress":"192.16.45.9",
                "subnetMask":"255.255.0.0",
                "gateWay":"255.255.255.255",
                "enable":"true"
            }
        ]
    },
    "receiveVideo2110":
    [ // this part is to receive a video feed where you set the connector,
the source address
      // and its port for both the connectors.
        {
            "stream":"video1",
            ...
        },
        {
            "stream":"video2",
            ...
        }
    ],
    "receiveAudio2110":
    [ // this part is to receive an audio feed where you set the connector,
the source address
      // and its port for both the connectors.
        {
            "stream":"audio1",
            ...
        },
        {
            "stream":"audio2",
            ...
        }
```

```
    ],
    "receiveAnc2110":
    [  // this part is to receive the ancillary data where you set the
connector, the source address
       // and its port for both the connectors.
        {
            "stream":"anc1",
            ...
        },
        {
            "stream":"anc2",
            ...
        }
    ],
    "transmitVideo2110":
    [  // this part is to send a video signal where you set the connector,
the destination address
       // and its port for both the connectors.
        {
            "stream":"video3",
            ...
        },
        {
            "stream":"video4",
            ...
        }
    ],
    "transmitAudio2110":
    [  // this part is to send an audio signal where you set the connector,
the destination address
       // and its port for both the connectors.
        {
            "stream":"audio3",
            ...
        },
        {
            "stream":"audio4",
            ....
        }
    ],
    "transmitAnc2110":
    [  // this part is to send ancilary data where you set the connector, the
destination address
       // and its port for both the connectors.
        {
            "stream":"anc3",
            ...
        },
        {
            "stream":"anc4",
            ...
```

```
        }
    ]
}
```

Sample Configuration File

You can download this sample JSON File.

Assign Configuration to PLAYDECK

The Configuration will be loaded while starting PLAYDECK and will be assigned to all Inputs and Outputs (if assigned).

For INPUTS, edit the full path to your JSON File in this Registry Key:
HKEY_CURRENT_USER\Software\Medialooks\MFormats\MFLive\AJA2\aja.ip_config

For OUTPUTS, edit the full path to your JSON File in this Registry Key:
HKEY_CURRENT_USER\Software\Medialooks\MFormats\MFRenderer\AJA2\aja.ip_config

---

Deltacst 2110 Cards

PLAYDECK allows to assign a network configuration file, which will be loaded upon PLAYDECK start.

Structure of Configuration File (JSON)

```
{
    "tx2110Setup":
    {
        "ethPort":0,
        "DHCP":"true",
        "ipAddress":"127.0.0.1",
        "subnetMask":"255.255.255.0",
        "gateWay":"127.0.0.1",
        "ipAddressMulticast":"239.1.32.32",
        "udpPort":16
    },

    "rx2110Setup":
    {
        "ethPort":0,
        "DHCP":"true",
        "ipAddress":"127.0.0.1",
        "subnetMask":"255.255.255.0",
        "gateWay":"127.0.0.1",
        "ipAddressMulticast":"239.1.32.32",
        "udpPort":16
    }
}
```

Sample Configuration File

You can download this sample JSON File.

Assign Configuration to PLAYDECK

The Configuration will be loaded while starting PLAYDECK and will be assigned to all Inputs and Outputs (if assigned).

For INPUTS, edit the full path to your JSON File in this Registry Key:
HKEY_CURRENT_USER\Software\Medialooks\MFormats\MFLive\MFDeviceDC\dc.ip_config

For OUTPUTS, edit the full path to your JSON File in this Registry Key:
HKEY_CURRENT_USER\Software\Medialooks\MFormats\MFRenderer\DC\dc.ip_config

Setup Device Reference Signal (GenLock)

Introducing a Reference Signal will force the Output Hardware to send Frames based on the Reference Clock.

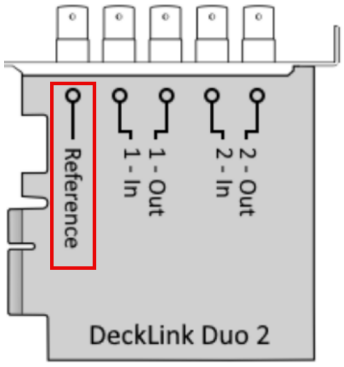
Multichannel SYNC as a use case

If you plan to synchronize multiple channel, you should connect an external clock to your device, then CUE all Channel via LINK (or Commands) and UNPAUSE all Channel. This will start all Channel (almost) SYNC, while the Reference Clock will make sure, that all Channel STAY SYNC. This is only guaranteed for all Channel, that output over the same Device.

This article shows, how to use Action Buttons to start multiple Channel in SYNC.

Providing a Reference Signal (GenLock)

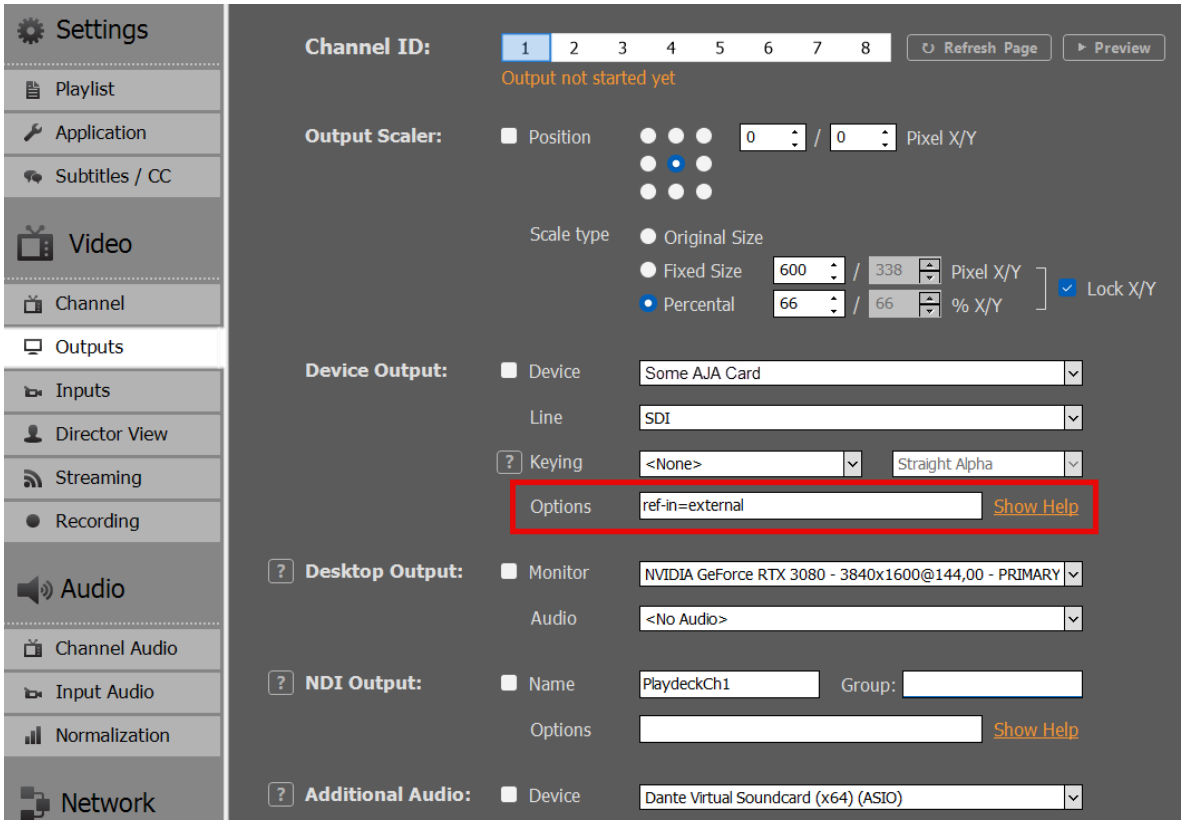
Please check with your manufacturer, if your Output Card supports a Reference Input. In this example we use “Blackmagic DeckLink Duo 2”:



If none of your other hardware generates a Clock Signal, you can use separate hardware to generate such a signal, like the Blackmagic Sync Generator.

Setting the Reference Input in PLAYDECK

Some Devices need specific reference input settings, like AJA Devices. In this Example, we set the AJA Reference Input to “external” for the Device:



If you click on SHOW HELP, you will see all other Options, that can be set for each device manufacturer.

Best Video Codecs for Playback

PLAYDECK can play almost any Codec out there. If the Codec is not supported by your GPU, it always falls back to CPU to make it work. There are very view exception: NotchLC for example cant be played with PLAYDECK.

We understand, that most of you have not much control over the Video Codec selection, as you receive your Video Files from Customer right before the show.

But if you have the Time to transcode the Video Files or even more are part of the production process, then you have more options than just H.264 MP4 Files.

We recommend going with AV1, if your NVIDIA supports it, except when you need Alpha-channel, than switch to HAP. In all other cases fall back to H.265/HEVC. ProRes has the best image quality, but is not really suited for real-time playback, as it is more of an editing and post-production codec.

There is also a Feature in PLAYDECK to quickly transcode your Video Files to a GPU-supported Codec. Please see this article.

H.265/HEVC

General-purpose codec. Use, if your NVIDIA GPU does not support AV1 and if you dont need Alpha-channel.

- **Developer**: JCT-VC
- **GPU Decoding:** Yes, but depends on NVIDIA GPU, see this List
- **Quality:** Good quality at all bitrates
- **Alpha-channel:** No
- **Features:** HDR, 10-bit color, 8K

AV1

Most efficient Codec. Use, if you your GPU supports it and you dont need Alpha-channel. Near-lossless quality and significantly smaller file sizes than ProRes.

- **Developer**: Alliance for Open Media (Open Source)
- **GPU Decoding:** Yes, but depends on NVIDIA GPU, see this List
- **Quality:** Better quality than H.265/HEVC, esp. at lower bitrates
- **Alpha-channel:** No
- **Features:** HDR, 10-bit color, 8K

HAP

Very universal Codec. Use, if you need Alpha-channel, and preferred over ProRes because of lower CPU load.

- **Developer**: Vidvox
- **GPU Decoding:** No, but lower CPU load than ProRes
- **Quality:** Better quality than H.265/HEVC, but less than AV1
- **Alpha-channel:** Yes (with HAP Alpha and HAP Q)
- **Features:**

ProRes

Offers highest image quality with ProRes 4444. Preserves fine details and colors exceptionally well, at the cost of high CPU load. Use only, if you can afford the CPU load and need highest possible visual quality. This is an editing codec and no real-time codec.

- **Developer**: Apple
- **GPU Decoding:** No. High CPU load
- **Quality:** Better quality than all other, esp. at higher bitrates
- **Alpha-channel:** Yes (with ProRes 4444)
- **Features:** HDR, 10/12-bit color, 8K

Transcode Clips for GPU Decoding

For a general overview of best possible Video Codecs for Playout, please see this article.

While PLAYDECK is able to transcode all Clips in realtime to your selected Output Format, this can put a heavy toll on your system during Playout, depending on your use case.

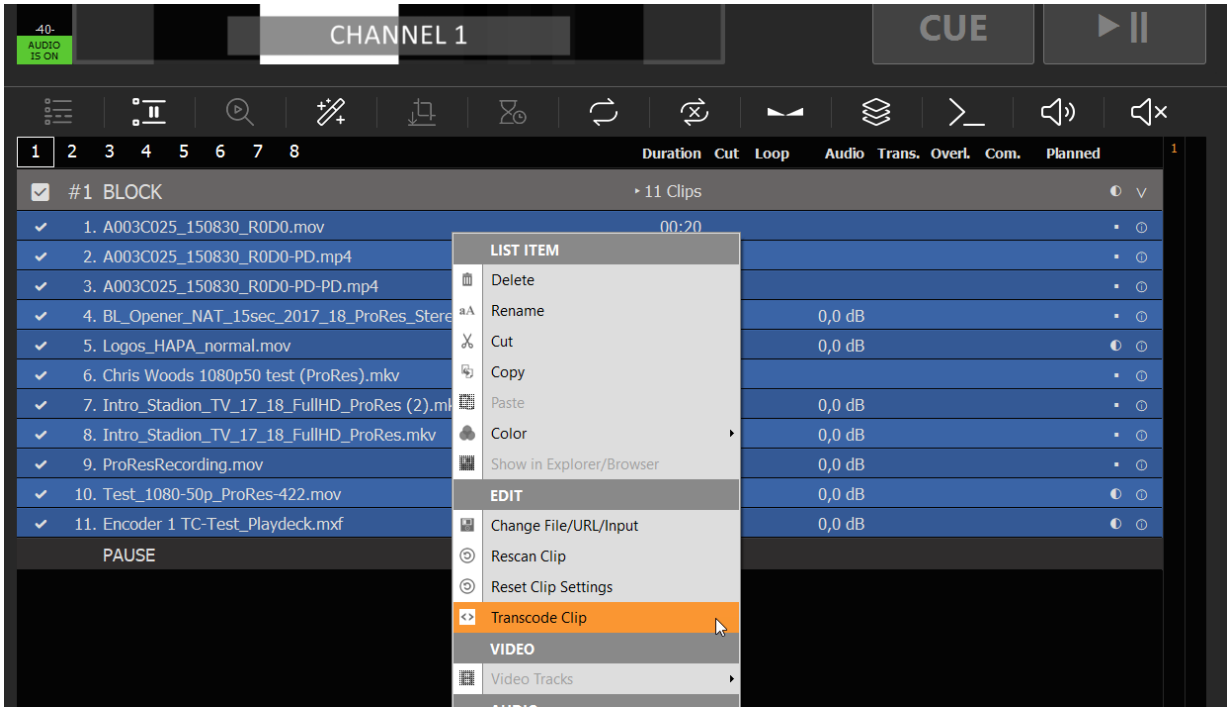
If your CPU usage reached critical level and you get playout lags (mostly perceived as audio stutter), you may want to make sure, that all Clips are decoded via the GPU, because most modern Graphic Cards can handle much more load than the CPU alone, thus making it possible to run multiple output channel with PLAYDECK.

Using the integrated transcoder

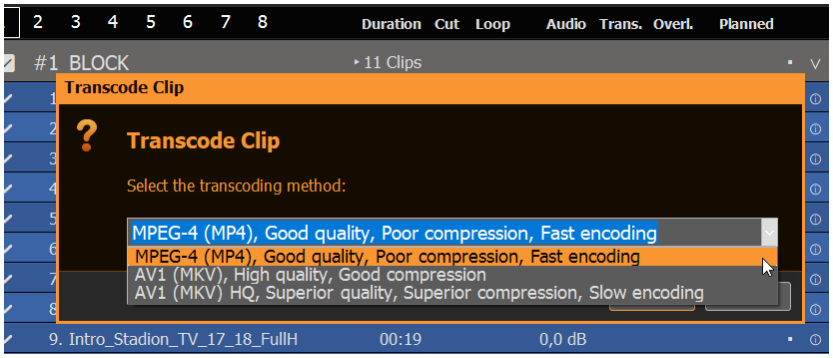
PLAYDECK has an integrated transcoder for video/audio files. It will make it more easy for you to QUICKLY transcode multiple files at once, e.g. if your show starts soon and you just don’t have the time to transcode them via 3rd party tool like Adobe Media Encoder.

In your example we have a Block of Files, that only can be decoded via CPU, like ProRes, and HAP-A Video Codec. We now select any Clip, then press CTRL+A to select all Clips in the Block. You could also use SHIFT to select a Clip range or just CTRL to select individual Clips.

Now we right Clip one of the selected Clips and select “Transcode Clips”:



You will not get a list of pre-defined target video codecs, which all support GPU decoding:



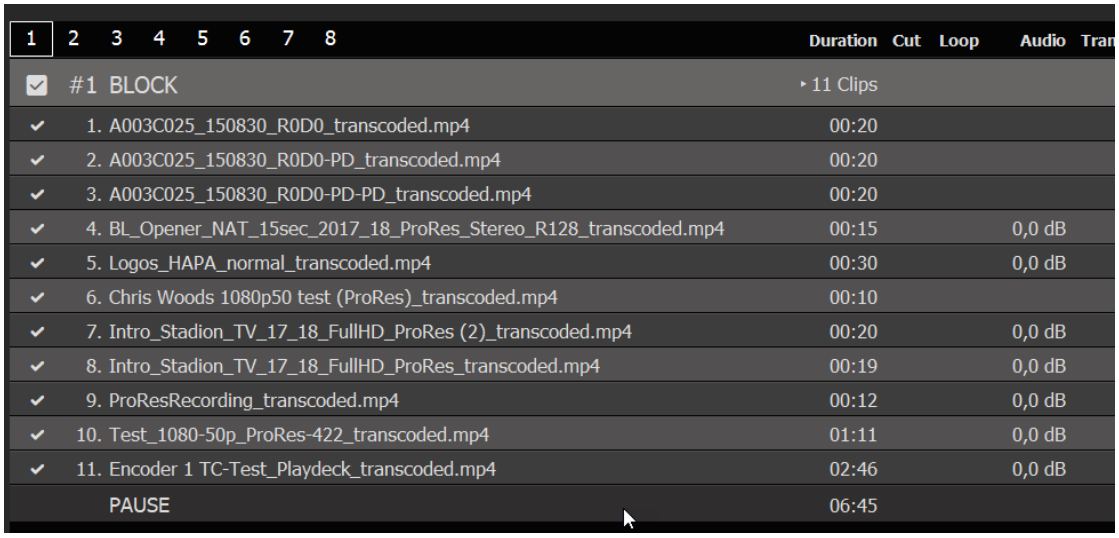
If you are in a hurry or have a huge number of clips to transcode, pick the first option “MPEG-4”, as this options gives you incredibly fast results, while still maintaining a good quality. In all other cases go with the 2nd option “AV1”, as this produces fairly small files on SSD/HDD, while producing extrem high quality files, and also support HDR. The last option is unnervingly slow, but also takes quality and compression a nodge higher, but the time/quality ratio is bad in this case. Use this for overnight-transcodings.

Note: Sadly, we can not offer transcoding in H.264 or H.265 because of license reasons.

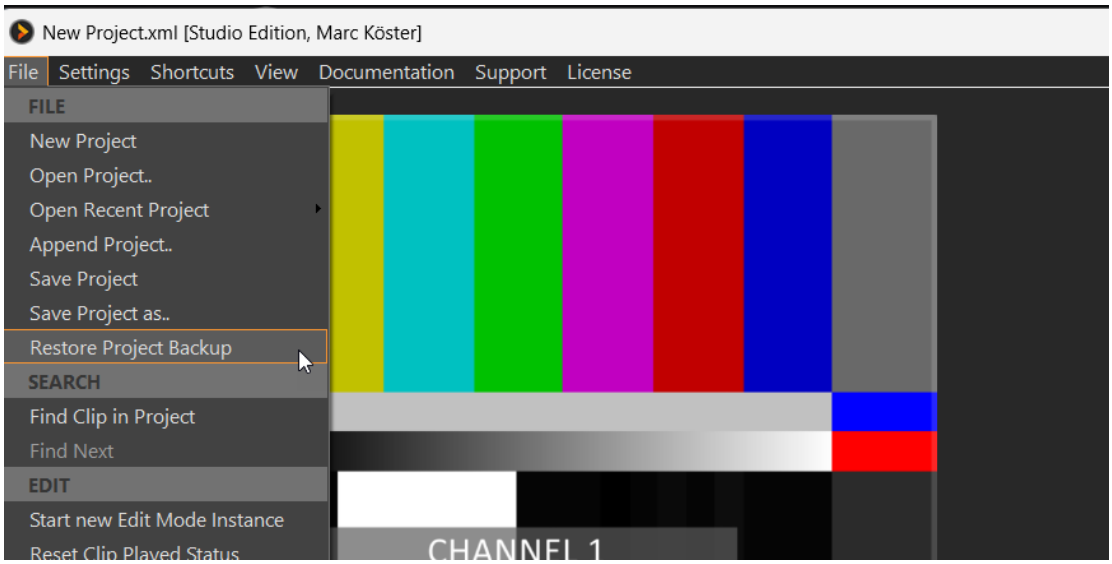
The transcoder will save the new file next to the old file and add “_transcoded” to the filename:

A003C025_150830_R0D0	00:00:20		13.10.2015 16:41:12
A003C025_150830_R0D0_transcoded	00:00:20		20.05.2025 03:17:35
A003C025_150830_R0D0-PD	00:00:20		12.04.2025 10:23:06
A003C025_150830_R0D0-PD_transcoded	00:00:20		20.05.2025 03:17:36
A003C025_150830_R0D0-PD-PD	00:00:20		12.04.2025 10:23:34
A003C025_150830_R0D0-PD-PD_transcoded	00:00:20		20.05.2025 03:17:37
BL_Opener_NAT_15sec_2017_18_ProRes_Stereo_R128	00:00:15		26.06.2017 10:42:59
BL_Opener_NAT_15sec_2017_18_ProRes_Stereo_R128_transcoded	00:00:15		20.05.2025 03:17:37
Chris Woods 1080p50 test (ProRes)	00:00:10		08.05.2025 15:20:03
Chris Woods 1080p50 test (ProRes)_transcoded	00:00:10		20.05.2025 03:17:45
Intro_Stadion_TV_17_18_FullHD_ProRes (2)	00:00:19		08.05.2025 14:49:22
Intro_Stadion_TV_17_18_FullHD_ProRes (2)_transcoded	00:00:20		20.05.2025 03:17:46
Intro_Stadion_TV_17_18_FullHD_ProRes	00:00:19		02.05.2025 01:51:10
Intro_Stadion_TV_17_18_FullHD_ProRes_transcoded	00:00:19		20.05.2025 03:17:48
ProResRecording	00:00:13		21.02.2021 23:35:00
ProResRecording_transcoded	00:00:13		20.05.2025 03:17:48
Test_1080-50p_ProRes-422	00:01:11		13.06.2019 11:17:40
Test_1080-50p_ProRes-422_transcoded	00:01:11		20.05.2025 03:17:50

The new Files will be automatically replaced and re-scanned in your Playlist:



If you need to at any time return to an earlier version of your Playlist, you can use the “Restore Project Backup” function:



What Files are being decoded on CPU in PLAYDECK

ProRes
This high quality video codec has it’s roots on MAC computers, as it also was developed by Apple. Sadly, Apple never released any codec informations for Windows Systems, so it’s still fairly impossible to decode ProRes via the GPU. If you need the Alpha Channel, we recommend using HAP-A video codec.

HAP and variants like HAP-A
These video codecs also support Alpha Channel and have very low CPU consumption.

DNxHD, MXF
The codecs also cant be decoded via GPU in PLAYDECK.

What GPU are recommended?

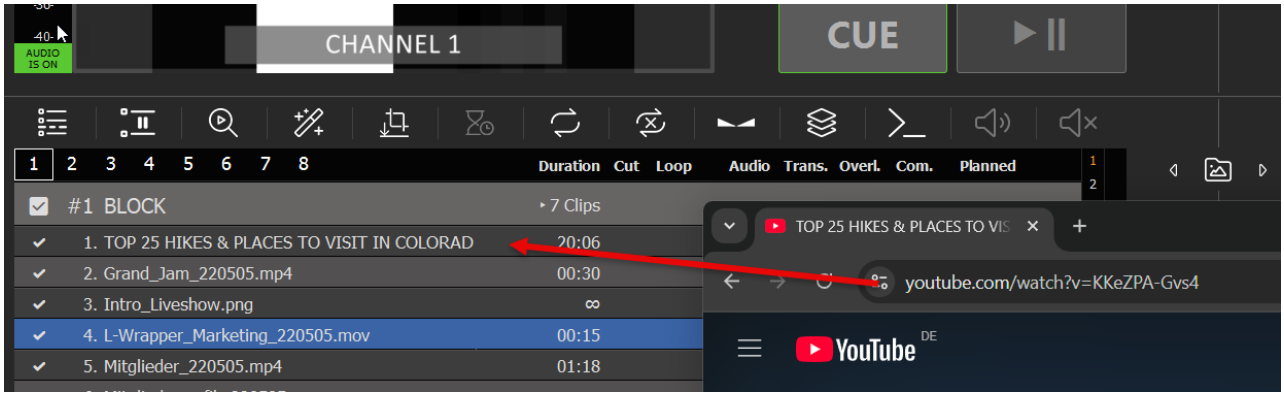
We refer to this article for PC Building for PLAYDECK.

Insert YouTube Videos into Playlists

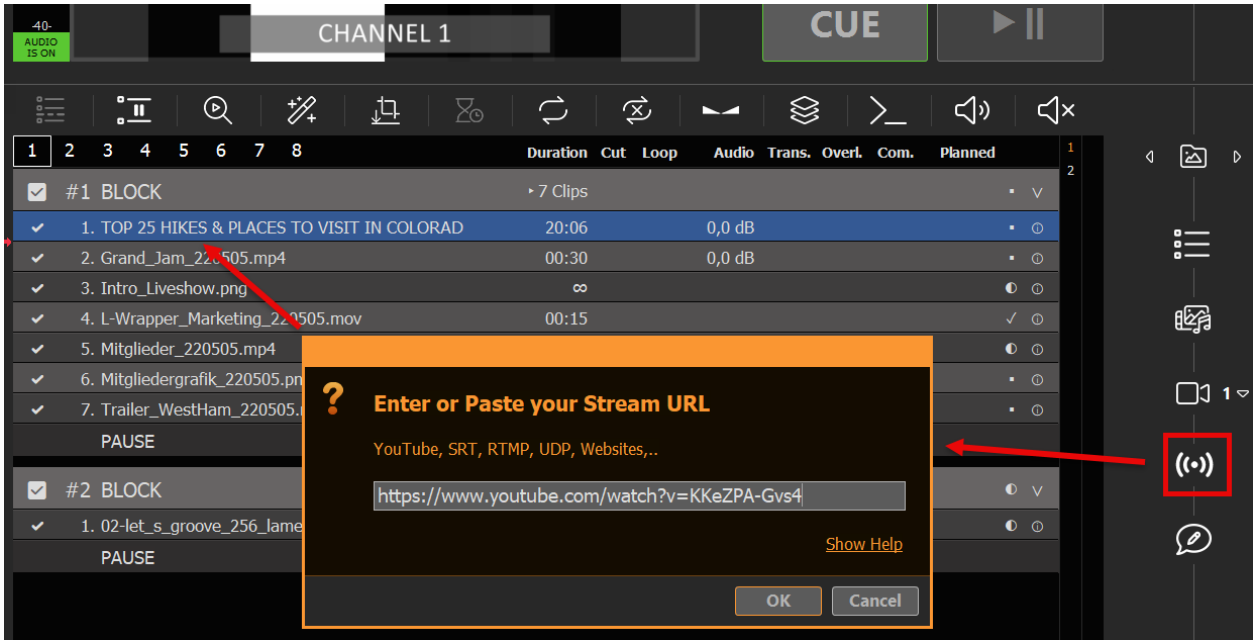
This article will show how to load and edit YouTube Videos.

Load YouTube into PLAYDECK

You can directly Drag Drop any YouTube URL from your Browser to PLAYDECK:



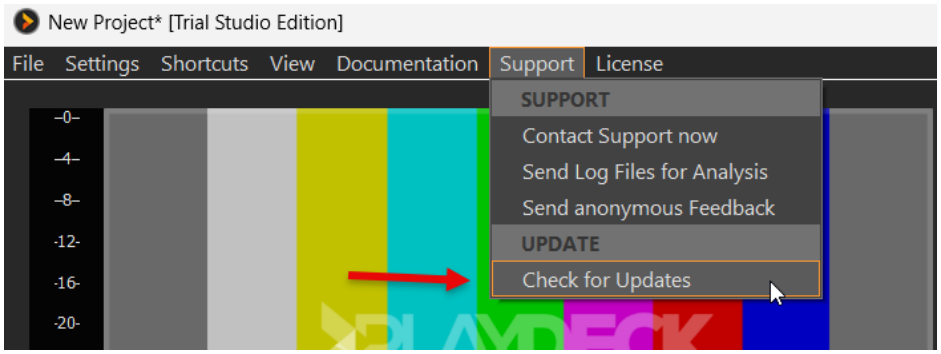
Or you can use the Stream Drag Drop Icon to paste/edit the URL:



Your YouTube Videos is ready to play out-of-the-box. The quality will be pre-selected dependend on your Playlist Vidoo Format and the Formats provided from YouTube.

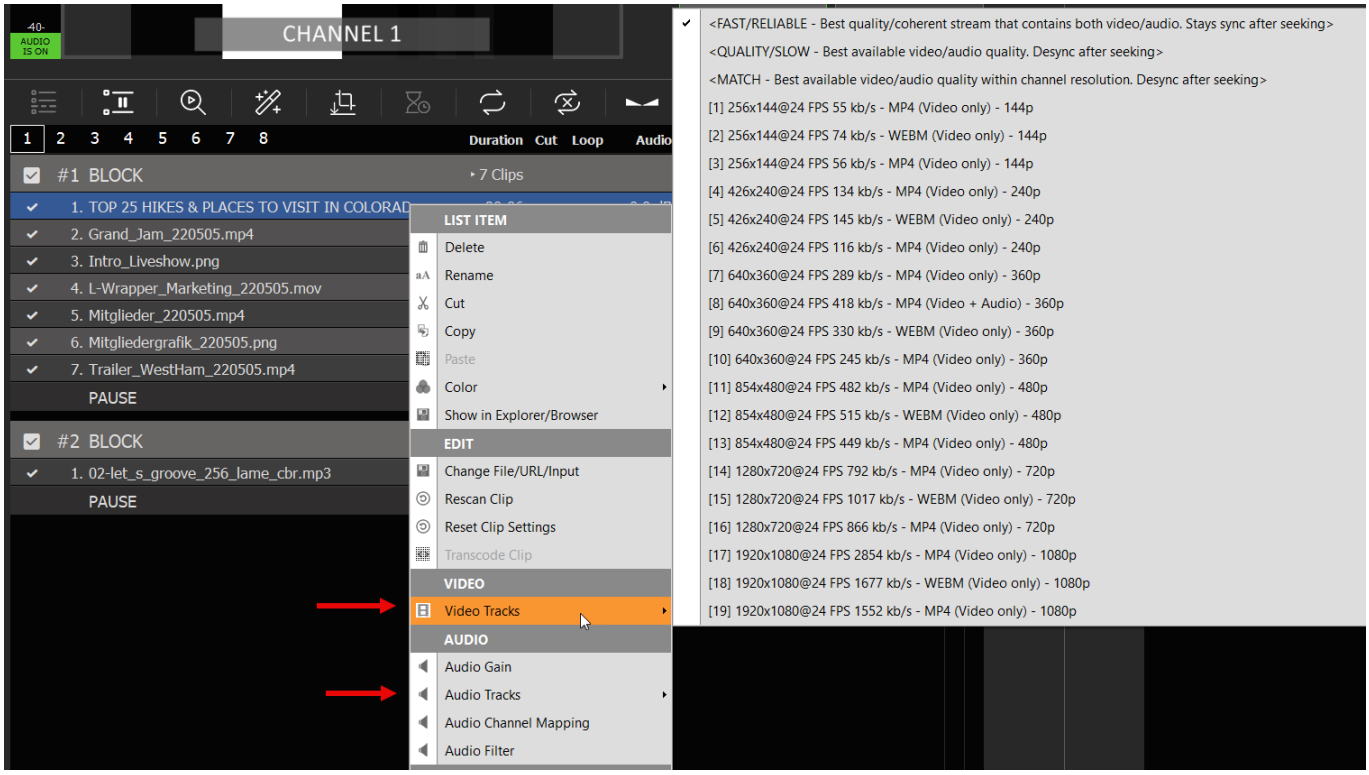
Update YouTube Driver

Should the added YouTube Clip be shown as UNPLAYABLE/RED in PLAYDECK, please update the integrated YouTube Driver. Chances are, YouTube made changes to its platform and you need a newer driver from us:



Change Quality, Video and Audio

YouTube Videos are always available in different qualities. You can switch Video Tracks by right-clicking the YouTube Video:



Note: YouTube provides combined Video+Audio Tracks, which are much faster. If you switch to non-combined Video Track, caching can take much longer and seeking can desync video/audio.

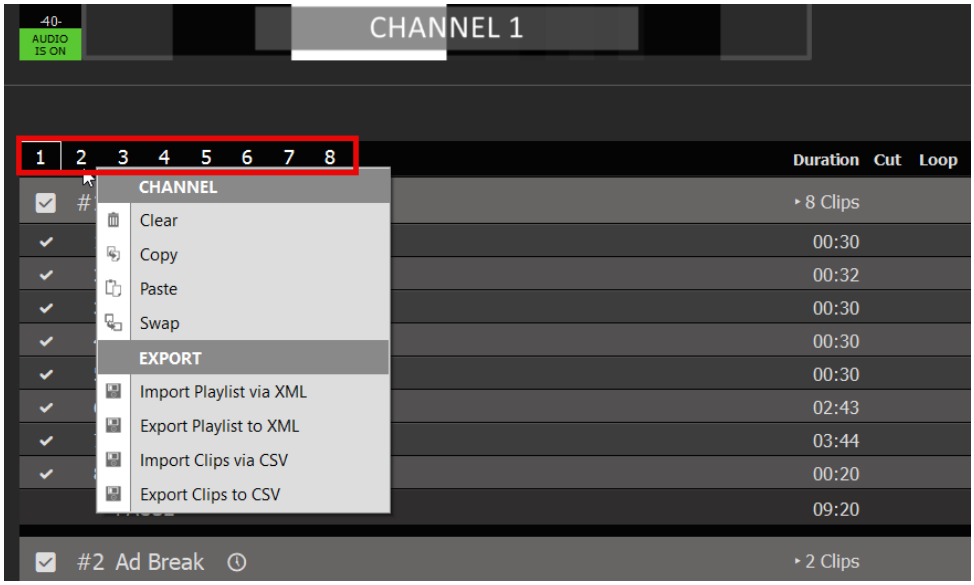
Prepare/Import Playlists externally

This article will show options to prepare your Playlists on another System or import Playlist from other Apps.

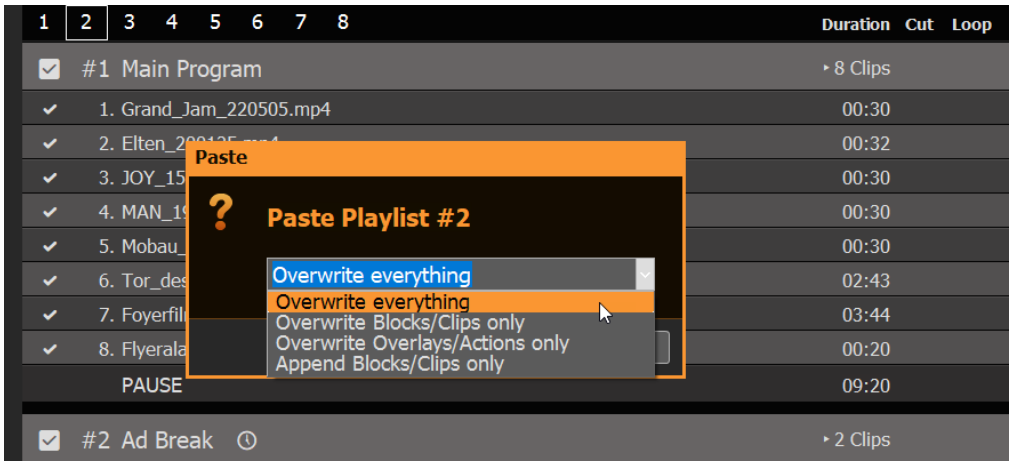
Use Free Channel

You have some spare Channel in PLAYDECK? Use them to prepare and test new Contents, then copy them to your Main Feed Channel. You can DRAG DROP Blocks from Channel to Channel, or copy them with pressing CTRL after starting DRAG DROP.

You can also copy, overwrite or append whole Playlists, by right-clicking the CHANNEL NUMBER:



If you paste new Content on an existing Playlist (or Import XML), you get the option to overwrite or append:



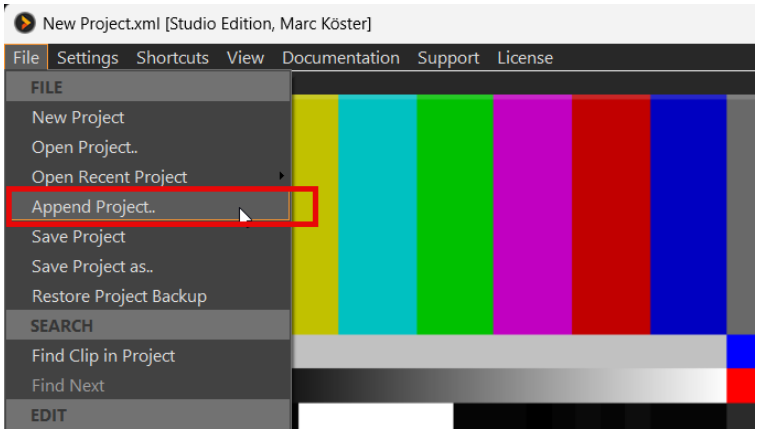
Use Playlist Trial Edition

You can install PLAYDECK on any other machine and edit your Playlists there. Except the Watermark of the Trial Edition, there are no other restrictions,

so you can prepare Playlists and send them to your Main System.

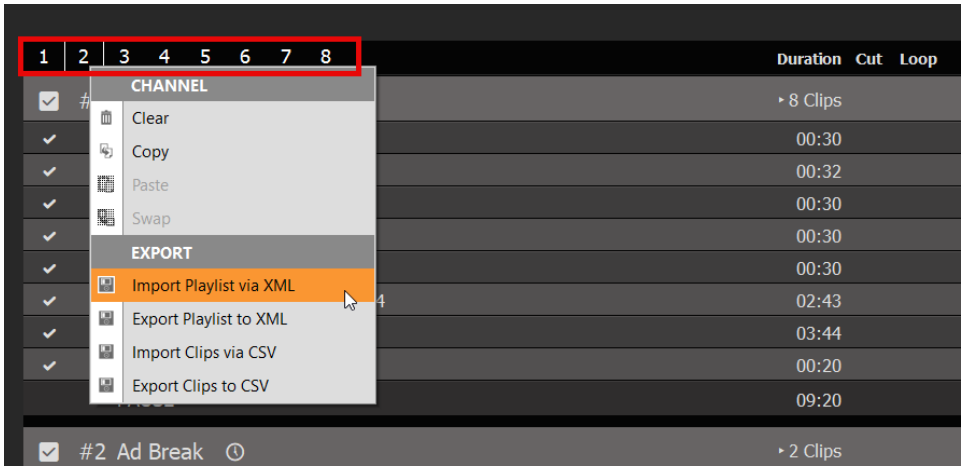
Append Playlist during Broadcast

You want to import new Playlists and don’t want to interrupt your Broadcast? No problem: Use the APPEND PROJECT Option in the FILE MENU:



This will load any PLAYDECK Project, but instead of replacing the Playlists, all Playlists will be extended by the Clips found in the Project. You can then safely remove any old Content.

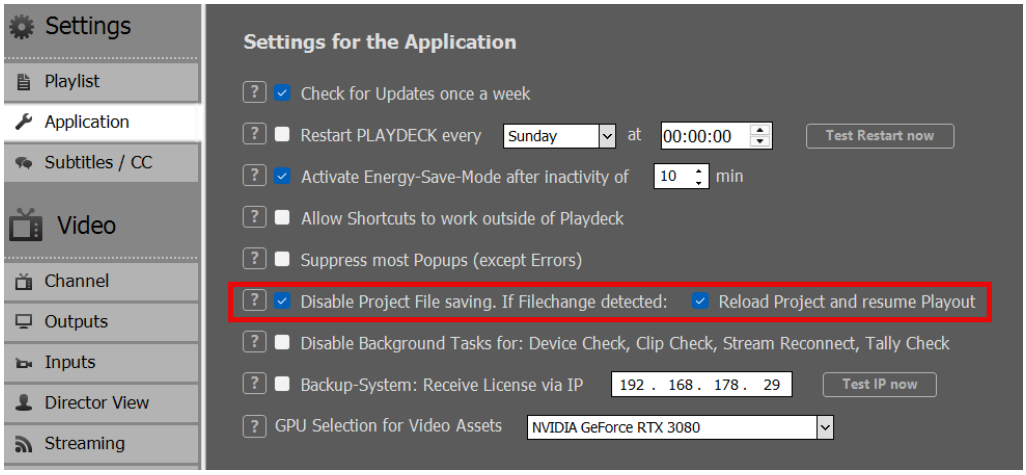
You can apply this procedure to individual Playlists by Exporting a Playlist to XML and Importing that XML on the other Machine. This is done by right-clicking the CHANNEL NUMBER:



This workflow is perfect for any day-to-day schedule, where you add new days and remove old ones.

Automatically Detect and Reload new Playlist

You can also OVERWRITE the current load Project File on your Main System and let PLAYDECK detect this and reload the Project, after which the Playout will resume. Enable this functionality in the Application Settings:



Note: RESUME PLAYOUT only works, if the current Playing Clip exists in the newly re-loaded Project, otherwise Playout stops for that Channel. The Clip is identified by a UNIQUE ID, so it could have moved to another Block in the re-loaded Project.

This workflow is best suited for any automated systems, where Project Files are generated automatically and uploaded to the Main System. Or when the Content Editor does not have access to the Main System and uploads the new Playlist to the NAS/Cloud Storage.

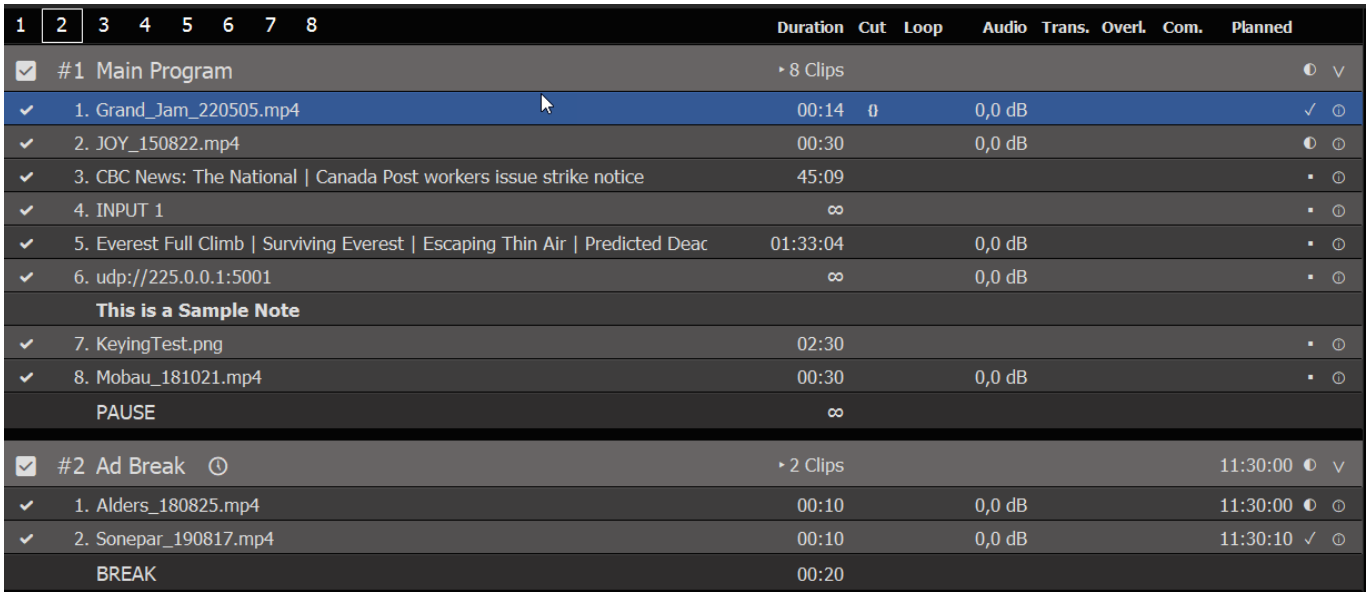
Interface with 3rd party Apps

We dont support the Playlist Format of other 3rd party Apps, as there is just too much diversity on the market, and alot Playlist Function (Mixing, Overlaps, Schedules) would not be compatible with PLAYDECK.

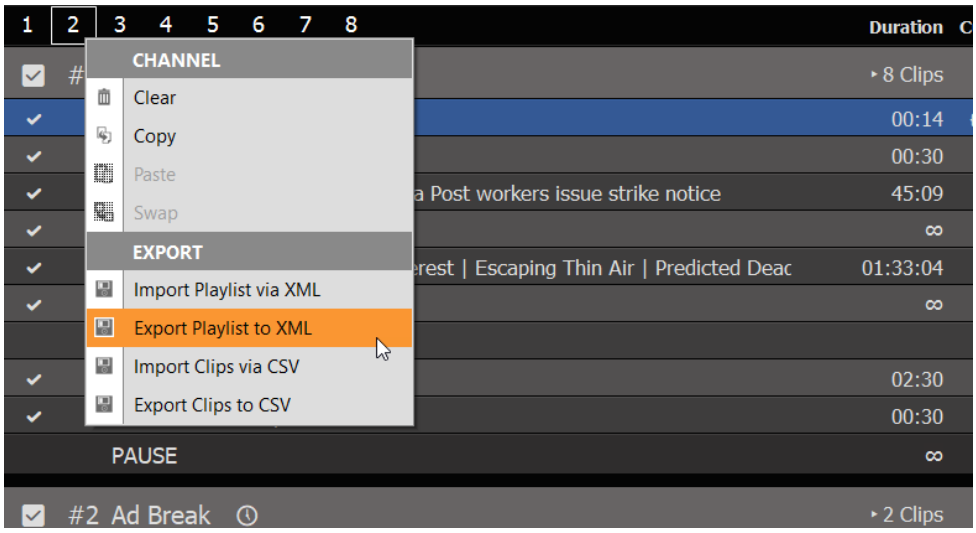
If you like to prepare your Playlist in a 3rd Party App, you need to export your Playlist to CSV, then Edit that CSV to work with PLAYDECK, then Import that CSV into PLAYDECK.

Why CSV? Because it is the most easy List Format, which support multiple values per Item. It can be edited with any Text or Table Editor, from Notepad to Excel.

Lets’s have a look at our CSV. For this we build ourself a Sample Playlists with 2 Blocks. We mixed in different Content Types: Video Clips with IN/OUT Points, Device Inputs, YouTube Videos, UDP Input Stream, GFX with Runtime, a Note. We also added a Schedule to the 2nd Block:



We now right-click the CHANNEL NUMBER 2 and select EXPORT TO CSV:



We then open the saved CSV File in Notepad (click here to download/open in new Tab):

```
Type;Active;Playable;Enumeration;Caption;Duration;Planned;Played;Filename;CutIn;CutOut;Schedule
Block;Y;Y;1.;Main Program;;;;;;;
File;Y;Y;1.;Grand_Jam_220505.mp4;00:14;;15:34:24;F:\Media\Ordner F\Grand_Jam_220505.mp4;9,04;23,96;
File;Y;Y;2.;JOY_150822.mp4;00:30;;23:54:42;F:\Media\Ordner E\JOY_150822.mp4;0;0;
YouTube;Y;Y;3.;CBC News: The National | Canada Post workers issue strike notice;45:09;;;https://www.youtube.com/watch?v=CP0JJOalE58;0;0;
Input1;Y;Y;4.;INPUT 1;;;;;0;0;
YouTube;Y;Y;5.;Everest Full Climb | Surviving Everest | Escaping Thin Air | Predicted Dead On Everest;01:33:04;;;https://www.youtube.com/watch?v=NoKAu_Amz7s;0;0;
Stream;Y;Y;6.;udp://225.0.0.1:5001;;;;udp://225.0.0.1:5001;0;0;
Note;;;;This is a Sample Note;;;;;;;
File;Y;Y;7.;KeyingTest.png;02:30;;;F:\Media\Testing\Alpha\KeyingTest.png;0;0;
File;Y;Y;8.;Mobau_181021.mp4;00:30;;;F:\Media\Ordner E\Mobau_181021.mp4;0;0;
Block;Y;Y;2.;Ad Break;00:20;11:30:00;;;;;21.05.2025 11:30:00
File;Y;Y;1.;Alders_180825.mp4;00:10;11:30:00;23:54:28;F:\Media\Ordner E\Alders_180825.mp4;0;0;
File;Y;Y;2.;Sonepar_190817.mp4;00:10;11:30:10;23:54:31;F:\Media\Ordner E\Sonepar_190817.mp4;0;0;
```

As you can see, its not “too” much Text Lines. Since the first Lines are just HEADER, it looks even more clear in a Table Editor:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Type	Active	Playable	Enumeration	Caption	Duration	Planned	Played	Filename	CutIn	CutOut	Schedule
2	Block	Y	Y	1.	Main Program							
3	File	Y	Y	1.	Grand_Jam_2205(	00:14		15:34:24	F:\Media\Ordner	9,04	23,96	
4	File	Y	Y	2.	JOY_150822.mp4	00:30		23:54:42	F:\Media\Ordner	0	0	
5	YouTube	Y	Y	3.	CBC News: The N	45:09			https://www.youtu	0	0	
6	Input1	Y	Y	4.	INPUT 1					0	0	
7	YouTube	Y	Y	5.	Everest Full Climb	01:33:04			https://www.youtu	0	0	
8	Stream	Y	Y	6.	udp://225.0.0.1:5001				udp://225.0.0.1:50	0	0	
9	Note				This is a Sample Note							
10	File	Y	Y	7.	KeyingTest.png	02:30			F:\Media\TestingV	0	0	
11	File	Y	Y	8.	Mobau_181021.m	00:30			F:\Media\Ordner	0	0	
12	Block	Y	Y	2.	Ad Break	00:20	11:30:00					21.05.2025 11:30:00
13	File	Y	Y	1.	Alders_180825.mp	00:10	11:30:00	23:54:28	F:\Media\Ordner	0	0	
14	File	Y	Y	2.	Sonepar_190817.	00:10	11:30:10	23:54:31	F:\Media\Ordner	0	0	

And this is how you write/generate CSV Files: You export your Content in any 3rd party app, open any editor and bring it in PLAYDECK FORMAT. But WHAT IS PLAYDECK FORMAT.

These are the REQUIRED COLUMNS. They can be it ANY POSITION (any column #):
Type = Block, File, YouTube, Input#, Stream, Note
Caption = Any Text to display in PLAYDECK
Filename = The Path/File or the URL (YouTube and Streams)

These are the OPTIONAL COLUMNS. They can be it ANY POSITION (any column #):
Active = The Checkbox in the first Column in PLAYDECK
Duration = Only used be File-Types that are Images. Leave empty or 0 for endless
CutIn / CutOut = IN- and/or OUT-Point for Trimming
Schedule = Only used by Block-Types

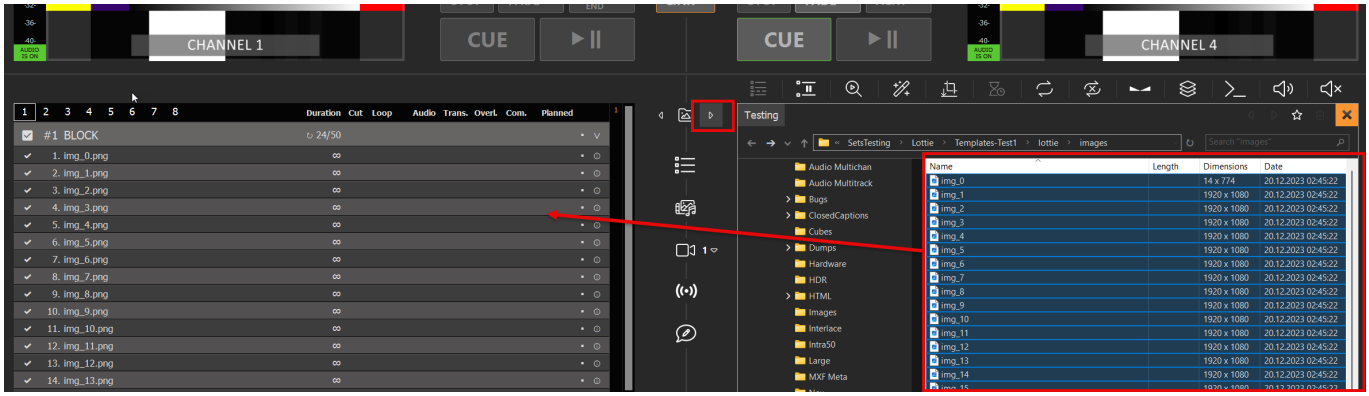
Note: More advanced Properties (e.g. Audio Gain) are not supported, as to keep our CSV Format clean and easy. To import/export ALL Properties, please use XML instead of CSV.

Creating a Diashow Single-Monitor

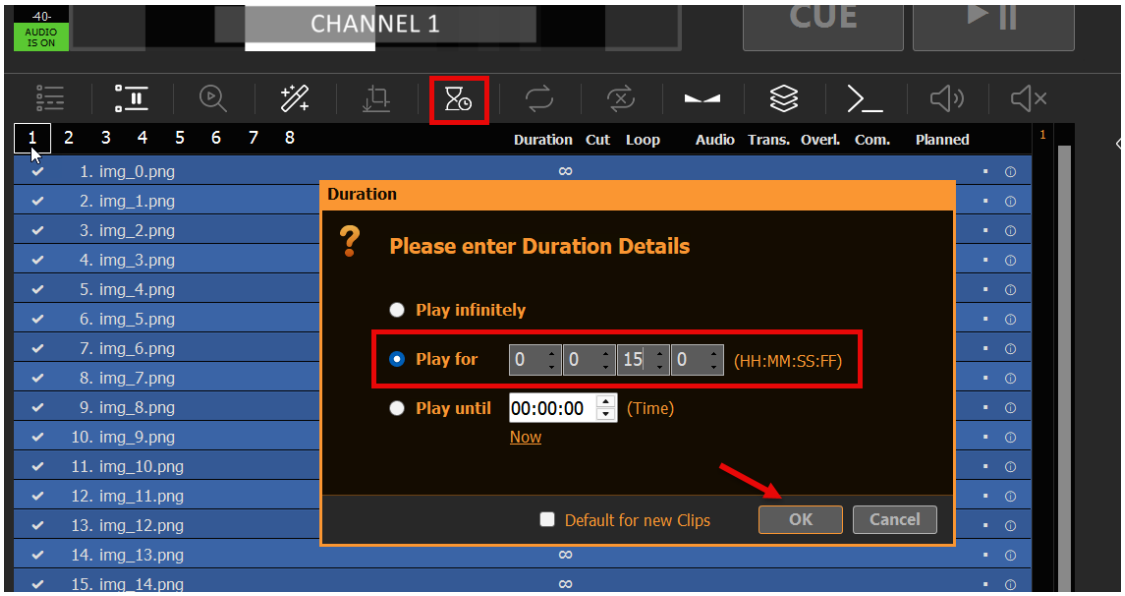
This article will show how to create a Diashow with Transitions and output on the same Monitor.

Creating the Show

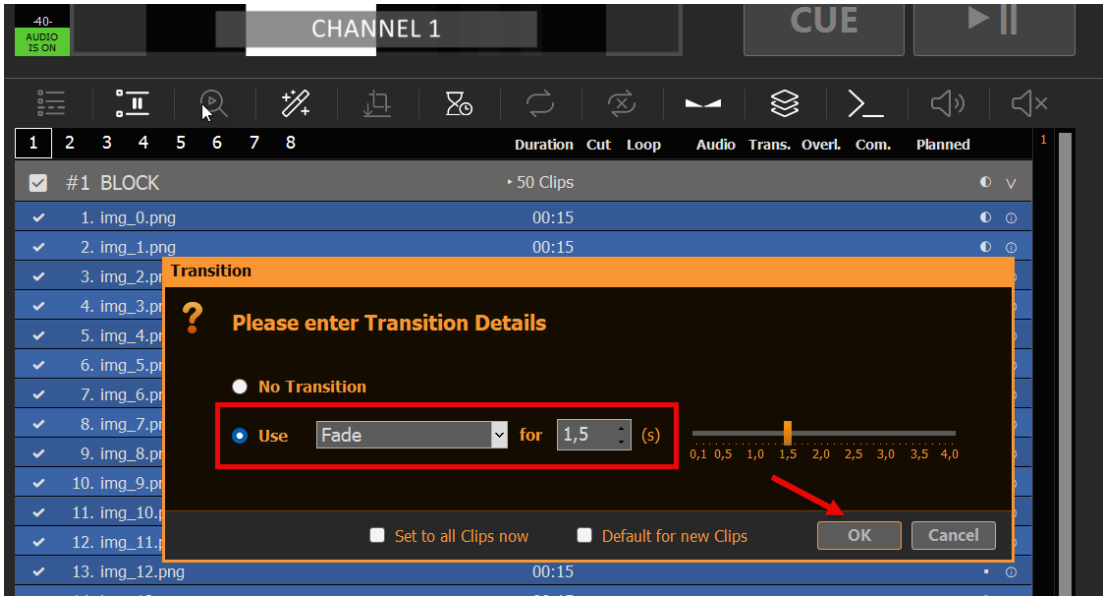
Open the integrated File Explorer by clicking the right-arror between the Playlists. The search for your images folder, select all images and add the to the Playlist by Drag Drop:



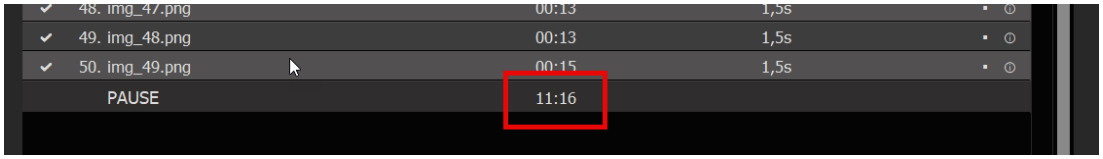
Select any Clip in the Playlist and press CTRL+A to select all Clips, then select the Icon for DURATION and enter a suitable duration for each image, e.g. 15 seconds:



While al Clips are still selected, click the Icon TRANSITION next and add any transition. In this case we use a smooth long Fade Transition:

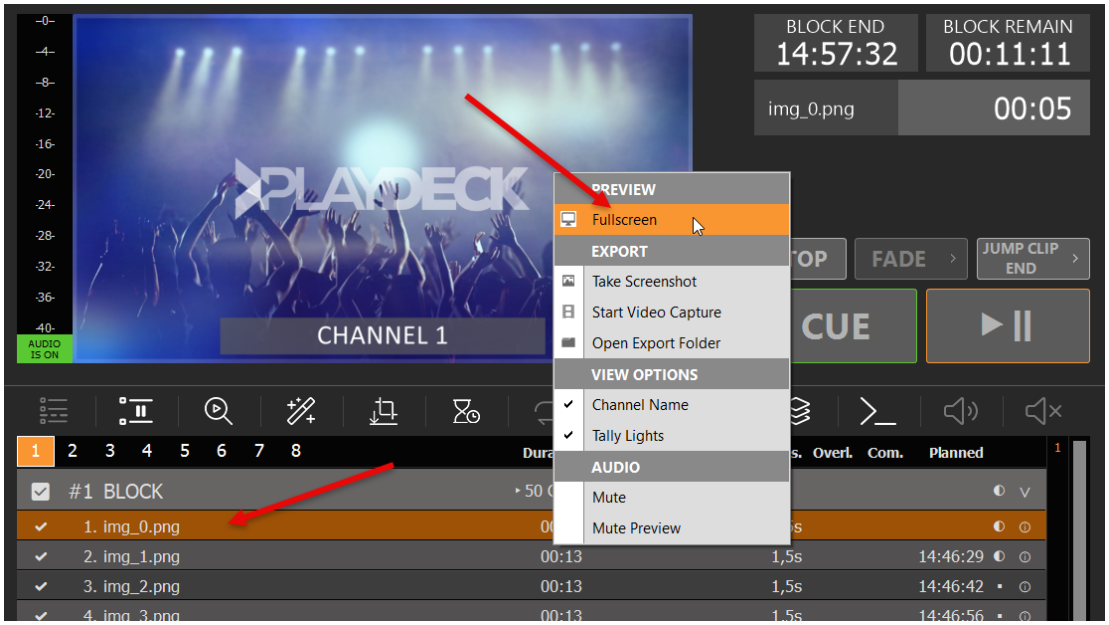


At the bottom of the Playlist you can see the total duration of your Diashow:



Playing the Diashow

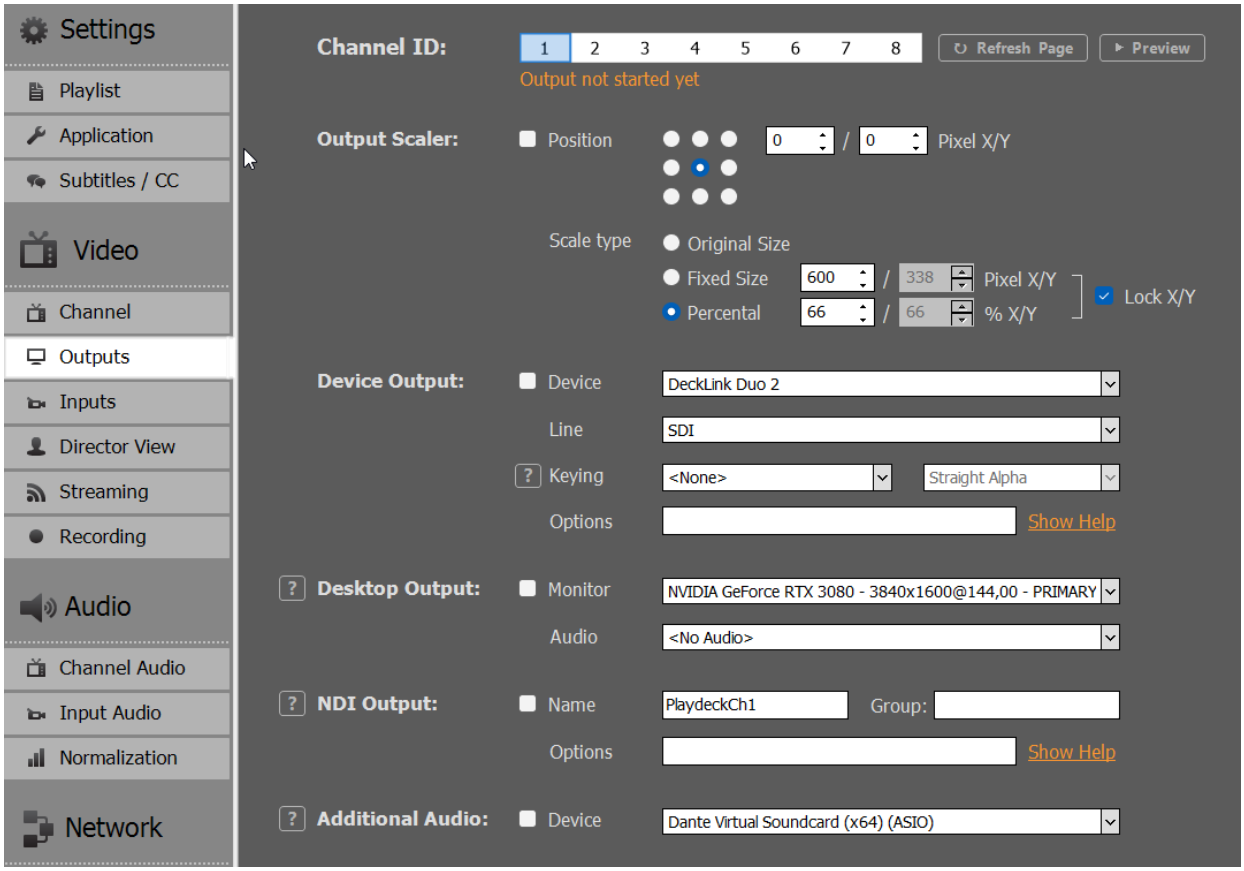
You simply double-click the first Clip (or use Buttons CUE+PLAY), then double-click the Preview Area (or right-click and select FULLSCREEN). Your Playout will now be maximized to your Monitor Size, at maximum Quality. And the Mouse Cursor will be hidden. Use ESC to exit Fullscreen.



Note: During Fullscreen, you can use the SPACE Key to pause/unpause Playback.

You can also output the video signal on a second monitor, or send it to your TV. You could also send it via NDI to another PC and open the signal with PLAYDECK (use the free Backup License), or VCL Mediaplayer or any other 3rd

party app:


Settings
Playlist
Application
Subtitles / CC
Video
Channel
Outputs
Inputs
Director View
Streaming
Recording
Audio
Channel Audio
Input Audio
Normalization
Network
Channel ID:
1 2 3 4 5 6 7 8
Refresh Page
Preview
Output not started yet
Output Scaler:
Position
0 / 0 Pixel X/Y
Scale type
Original Size
Fixed Size 600 / 338 Pixel X/Y
Percental 66 / 66 % X/Y
Lock X/Y
Device Output:
Device
DeckLink Duo 2
Line
SDI
Keying
<None>
Straight Alpha
Options
Show Help
Desktop Output:
Monitor
NVIDIA GeForce RTX 3080 - 3840x1600@144,00 - PRIMARY
Audio
<No Audio>
NDI Output:
Name
PlaydeckCh1
Group:
Options
Show Help
Additional Audio:
Device
Dante Virtual Soundcard (x64) (ASIO)

External Overlays with ClassX (and other)

This article will show how to trigger external Overlay Engine during your Playout.

Create external Overlay with ClassX

We use ClassX Liveboard as an example to show, how to work with external Overlays. This sample can be easily copied to any other solution, like: CasparCG, Singular.live.

ClassX Liveboard is a professional CG Playout Solution for automating dynamic overlay contents. You can register for a Demo Version on their website.

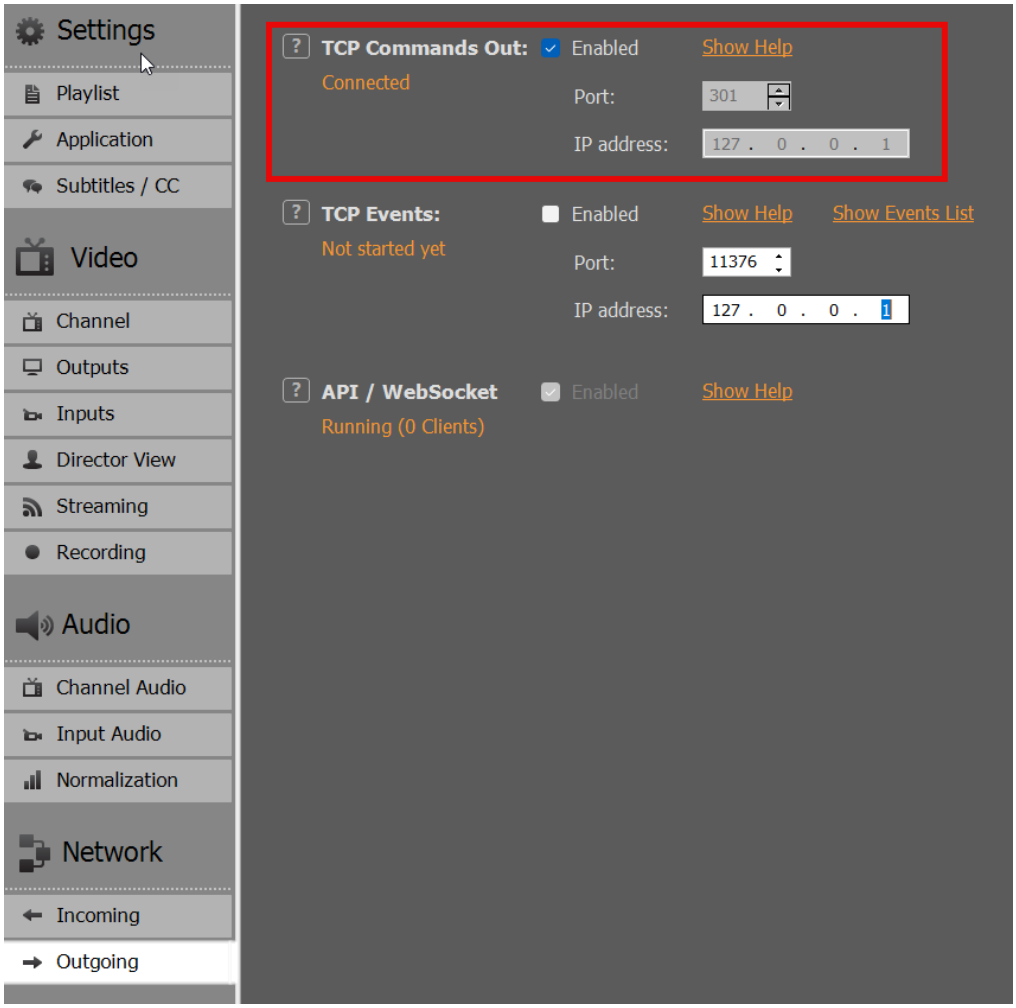
After installation of LiveBoard, you can simply add Videos, GFX and other Contents via the user interface. There you also name the Contents with a unique identifier, e.g. “MyLowerThird” for later reference.

LiveBoard allows to connect via TCP and send Commands to show/hide certain Contents. A complete list of Commands is always found in the Window installation path:
C:\Program Files\…\ClassX_Applications_v6\remotecontroltester\commands

For now, these 2 Commands are enough for us:

- LBC_PLAYCONTENT “MyLowerThird”
- LBC_STOPCONTENT “MyLowerThird”

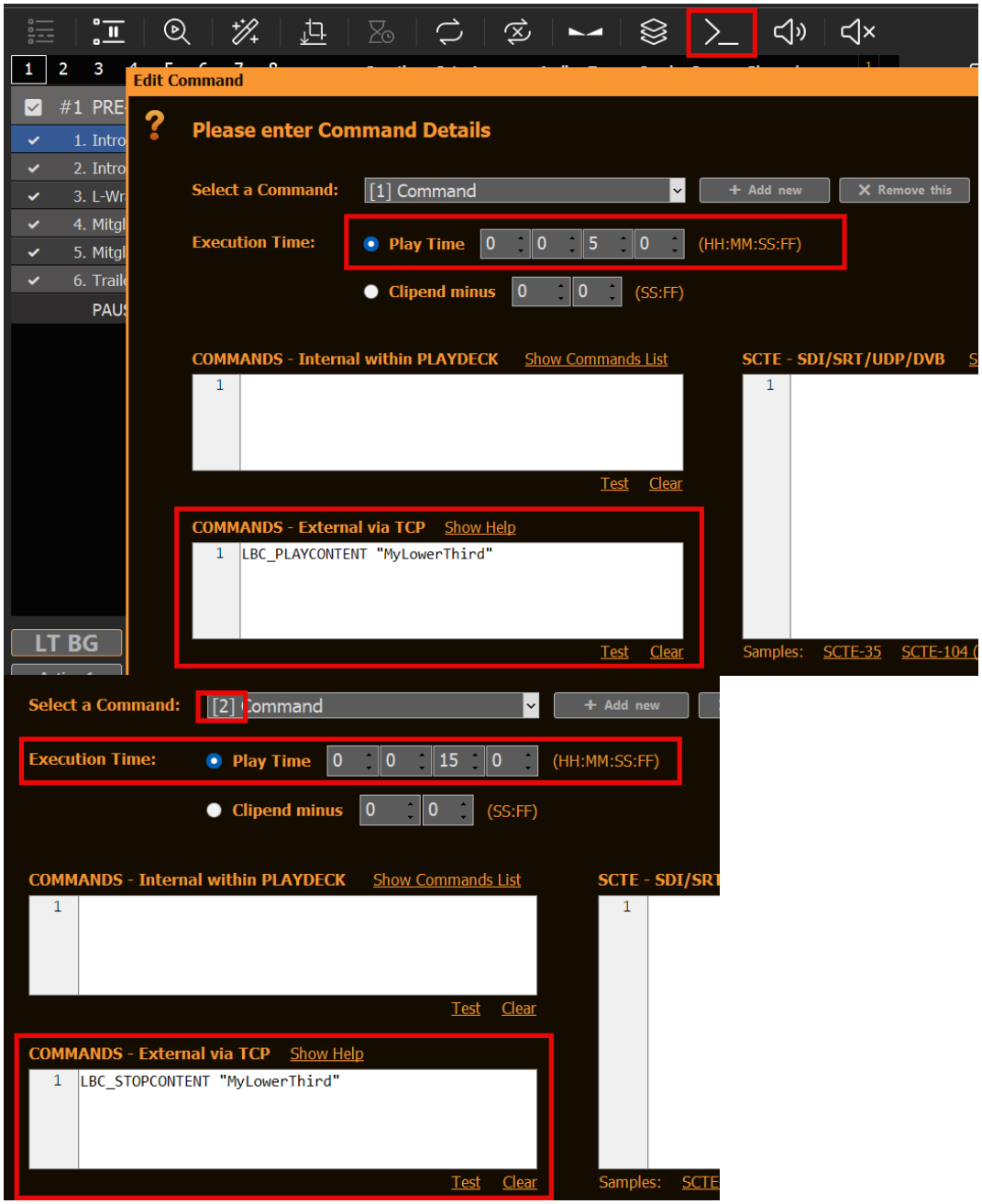
Lets enable our TCP in PLAYDECK to talk to LiveBoard. The default Port of LiveBoard is 301. Enabling this will show CONNECTED, if the LiveBoard Server is running. Otherwise PLAYDECK will try to connect every 5 seconds, so you can start LiveBoard before or after PLAYDECK:



You can now add the above Commands to:

- Clips
- Blocks
- Actions
- Other Overlays

For our Sample we want the external Overlay to show together with a specific Clip. We select this Clip in the Playlist and click the COMMANDS Icon, then add our 2 LiveBoard Commands from above. In this case with start the Overlay 5 seconds into the Clip and show it for 10 seconds in total:



Mixing Video Content and Overlays

If you dont use an external Mixer to mix the Overlays with Alpha Channel over your PLAYDECK playout, you can also use Internal Keying, where you feed the Overlay Signal into PLAYDECK, and have your Output Hardware mix both Layer. See this article on Internal Keying.

If you want to mix on a software level, you can use vMix on the same System as ClassX and PLAYDECK.

Dante Virtual Soundcard

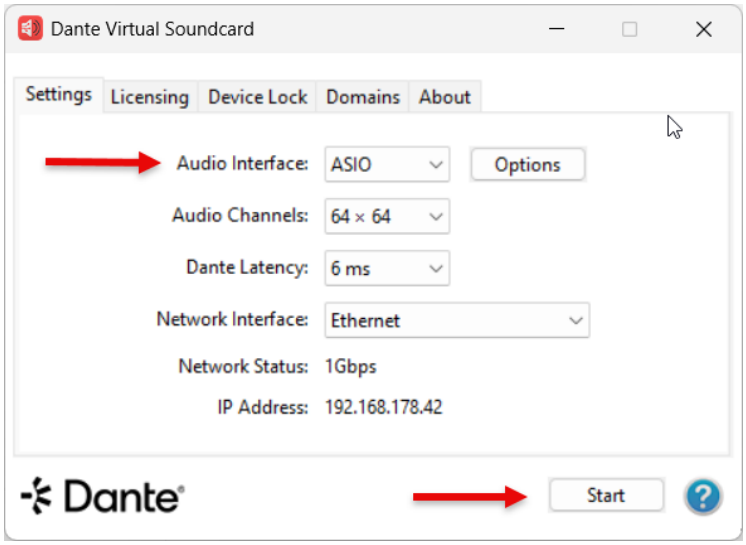
If you dont know DVS yet: It’s an Audio connection system for Ethernet (LAN).
https://www.getdante.com/products/software-essentials/dante-virtual-soundcard/

You can use DVS via **ASIO** or **WDM**:

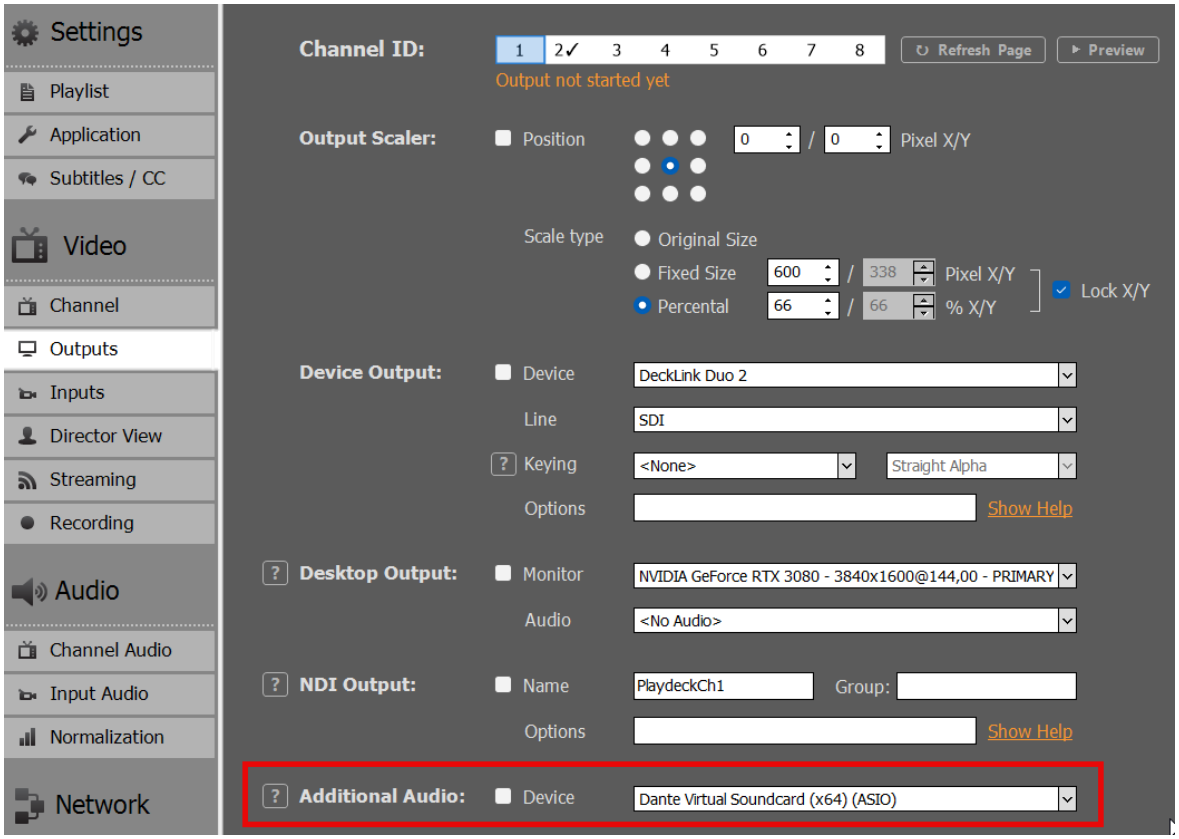
- With ASIO you can transmit up to 64 Audio Channel across all PLAYDECK output channel.
  It is the recommended method to use DVS with PLAYDECK.
- With WDM you can transmit up 16 Audio Channel with one Stero Pair per PLAYDECK output channel.

DVS via ASIO

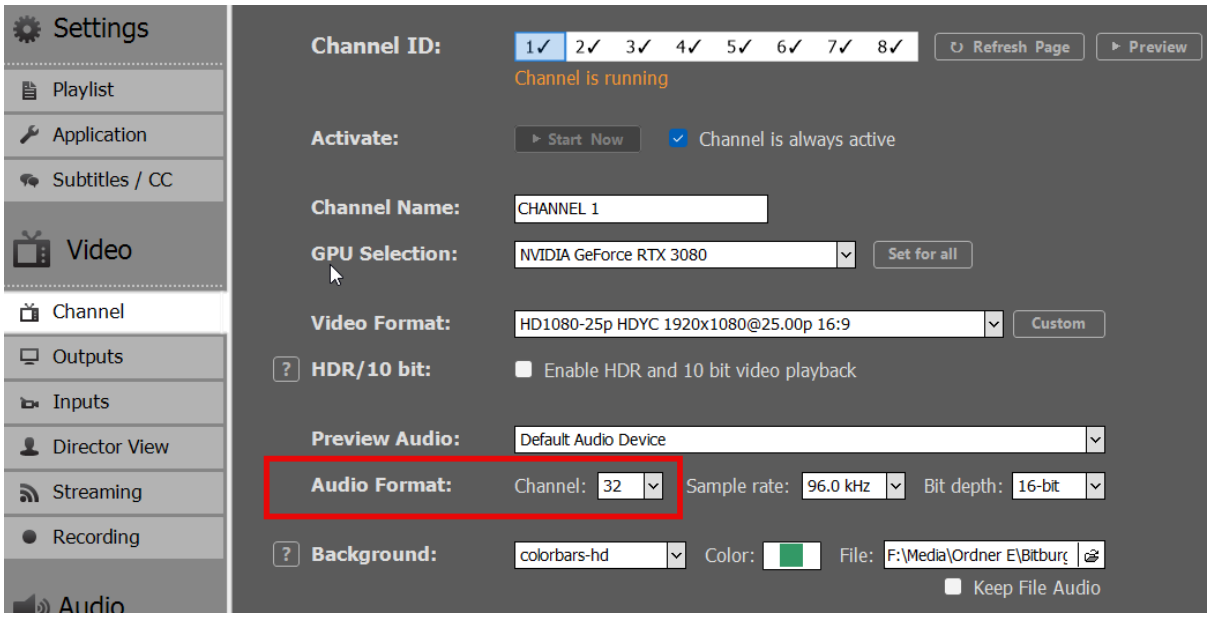
Start the ASIO Device by opening DVS, selecting ASIO and click START:



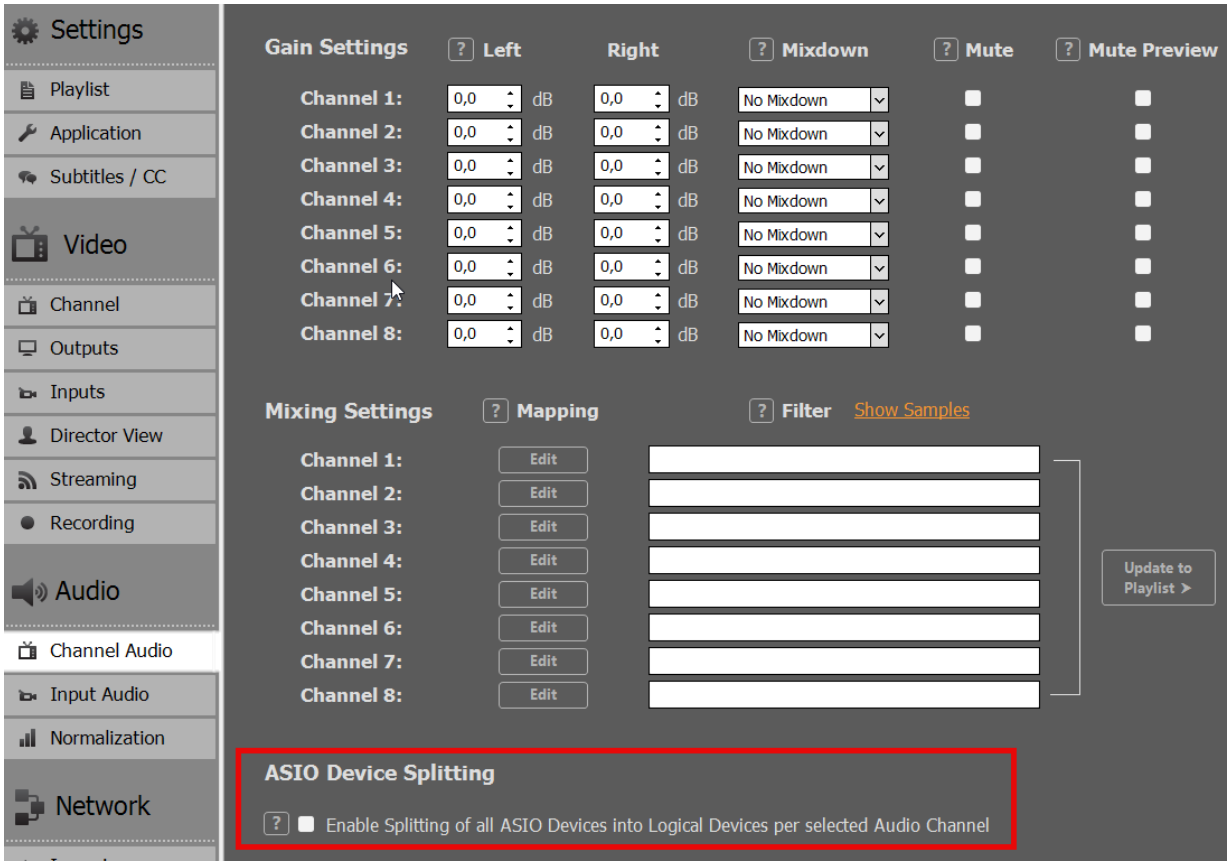
You can select the ASIO Device in PLAYDECK now:



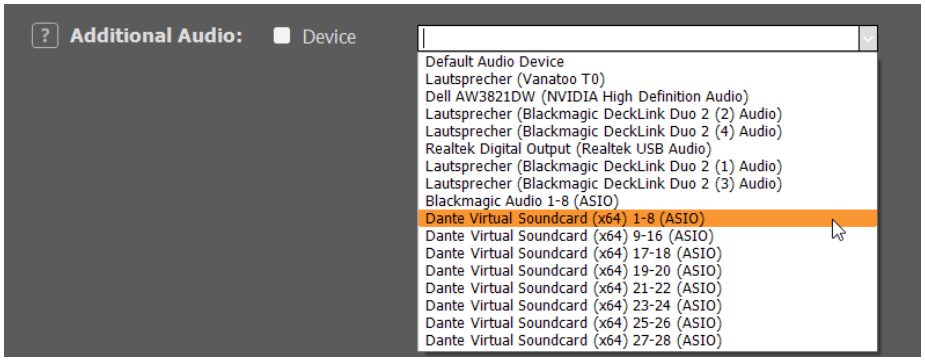
PLAYDECK will now start to send all Audio Channel you have selected for that Output Channel, which can be up to 32 Audio Channel:



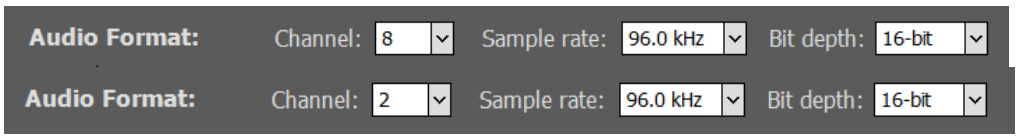
To use the ASIO Device for more than one Output Channel, you need to activate ASIO Device Splitting:



You will now receive multiple ASIO Devices, so you can assign one Device per PLAYDECK Output Channel:

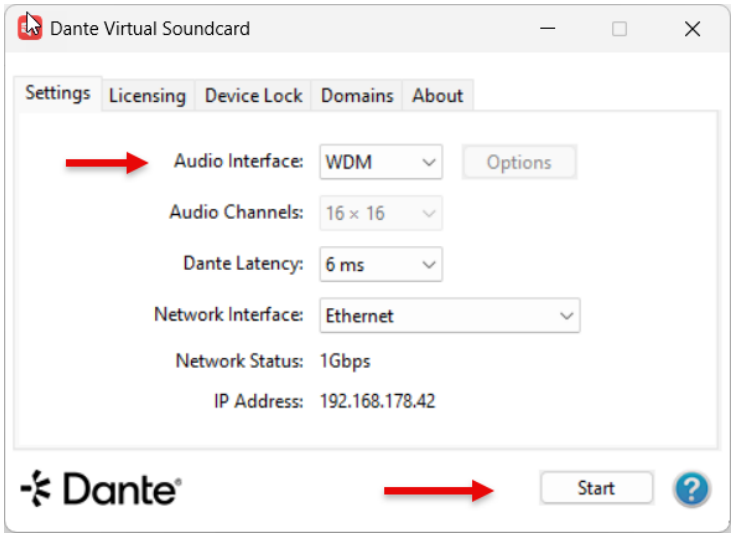


The Splitting will be done by the Audio Channel you selected per Output Channel. So in the above example, we have set PLAYDECK Output Channel 1 and 2 to 8 Audio Channel and all other Output Channel to 2 Audio Channel.



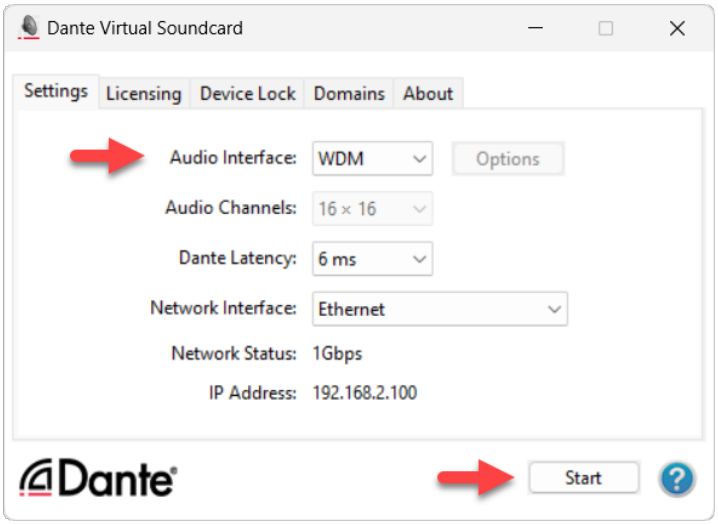
DVS via WDM

Start the WDM Driver by opening DVS, selecting WDM and clicking on START:

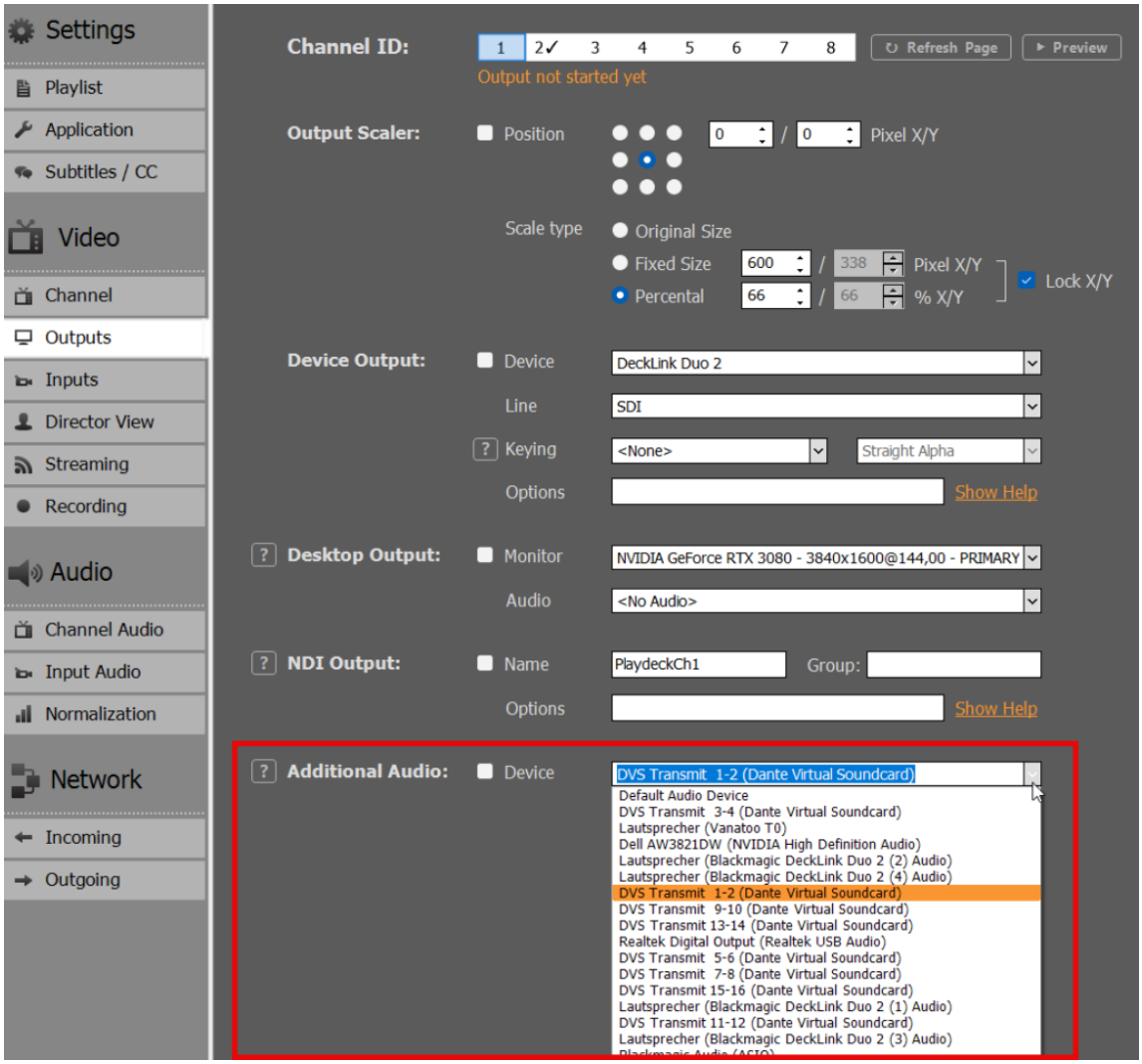


This is the product page:
https://www.audinate.com/products/software/dante-virtual-soundcard

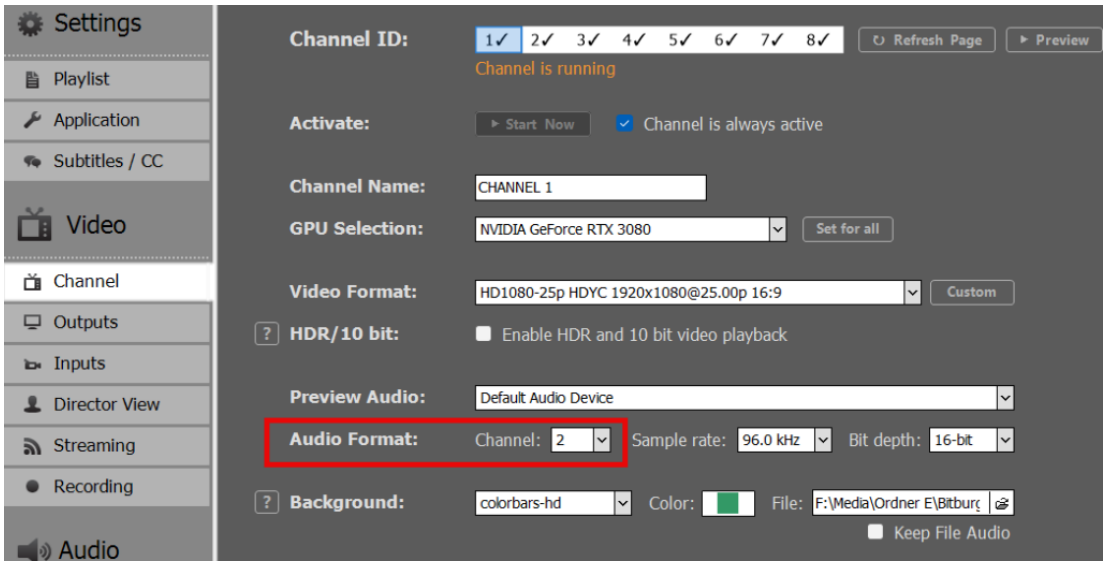
Once you installed DVS on the PLAYDECK machine, select WDM as Audio Interface and START:



You can now assign on DVS Audio Device (Stereo Pair) per PLAYDECK Output Channel:



Only the first to Audio Channel of that Output Channel are send, so make sure to have set your Audio Channel to 2:



Troubleshooting

Stuttering Video

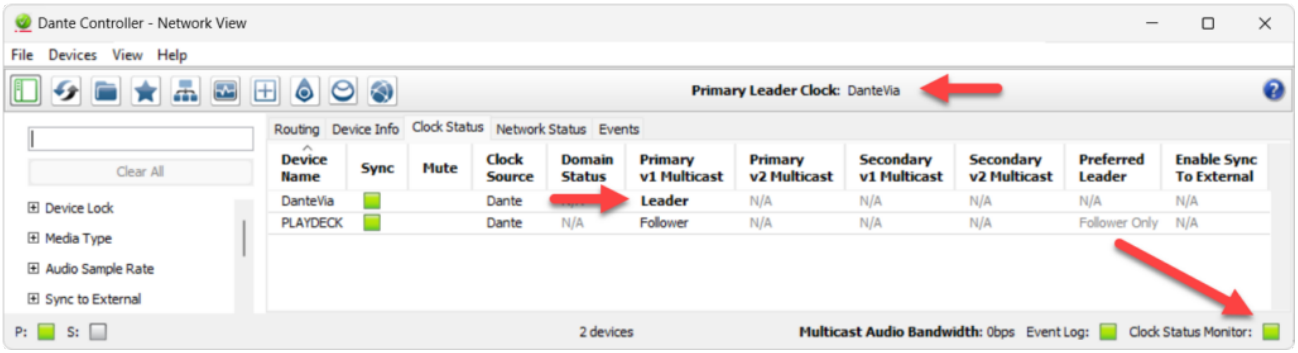
If you assigned DANTE to the Channel in PLAYDECK and your Playback does not start or stutters very slow: You need DANTE CLOCK in your Network.

The Playout in PLAYDECK will not continue, unless the DANTE CLOCK is triggering the Playout. This behaviour CAN NOT be avoided.

Most Audio Mixer, which support DVS, will have a Clock integrated. Maybe it needs to be activated. Please also watch this Official Video from Dante for Dante clocking.

You can also create an artificial DANTE CLOCK by installing DANTE VIA on a SECOND PC in the Network (cant be installed on the same System as PLAYDECK). Once installed, it will automatically designate itself as the Leader Clock and connect to the DVS by itself without further user intervention.

hat everything is working as expected. It is simply a controlling and reporting tool:

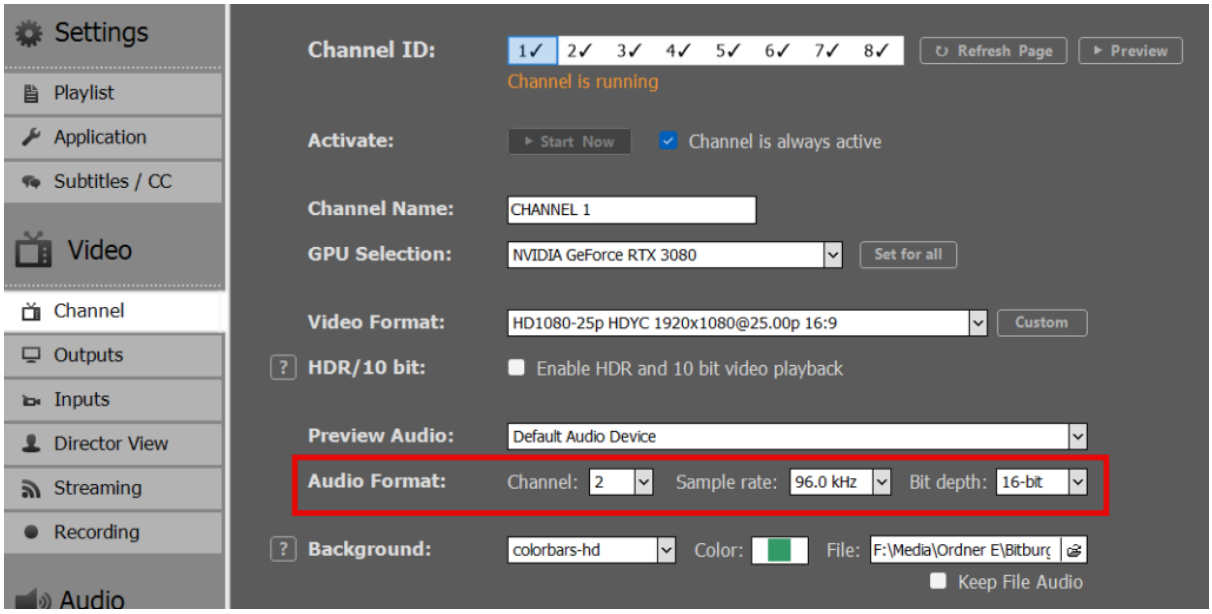


If everything is setup correctly, you will see a GREEN LIGHT in the bottom right corner, indicating that your DVS is Clock-enabled and ready to use in PLAYDECK.

Once you see this GREEN LIGHT, PLAYDECK will now play all Clips correctly.

Bad Audio Quality

This most likely happens with the WDM Driver and different Audio Settings. Make sure to adjust the PLAYDECK Channel Audio Setting and DVS Audio Device Setting to the same Format. This avoids transcoding auf Audio and will result in a higher overall Audio Quality.



Audio Channel Mixing and Routing

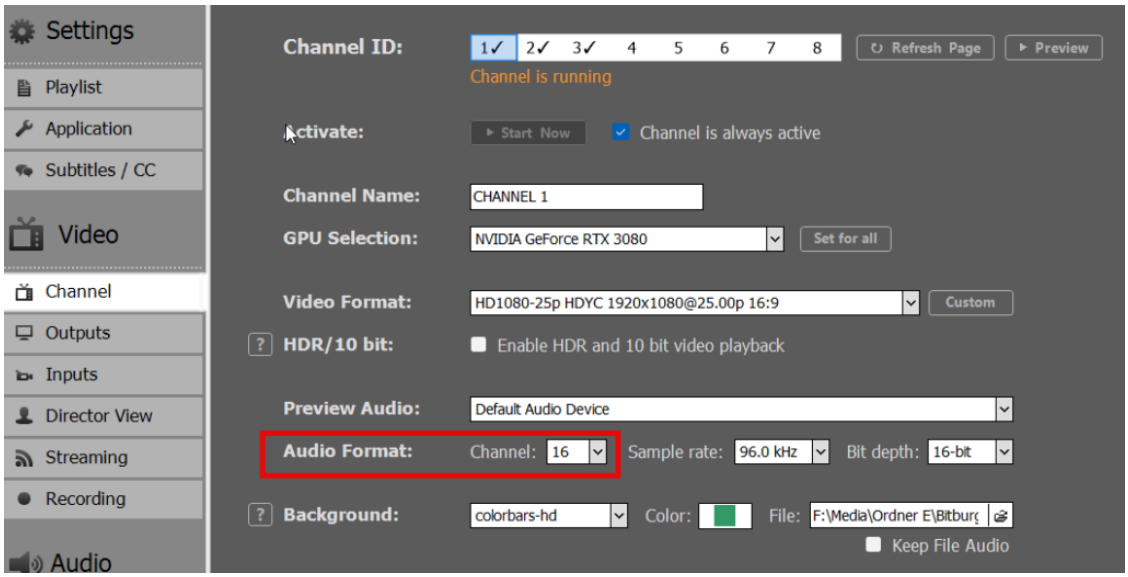
If you work with Multichannel Audio, you might have to mix down your Audio Channel. See this article for how to work with multichannel audio.

Multichannel Audio and Mixing

PLAYDECK support 32 Audio Channel per Output Channel. You can pass-thru Audio Channel (via SDI, NDI, Streams, ASIO), Mix-down to Stereo/Mono or Mix-up to Multichannel. Please note, that we have a sperate Post for Dante Virtual Soundcard.

Pass-thru

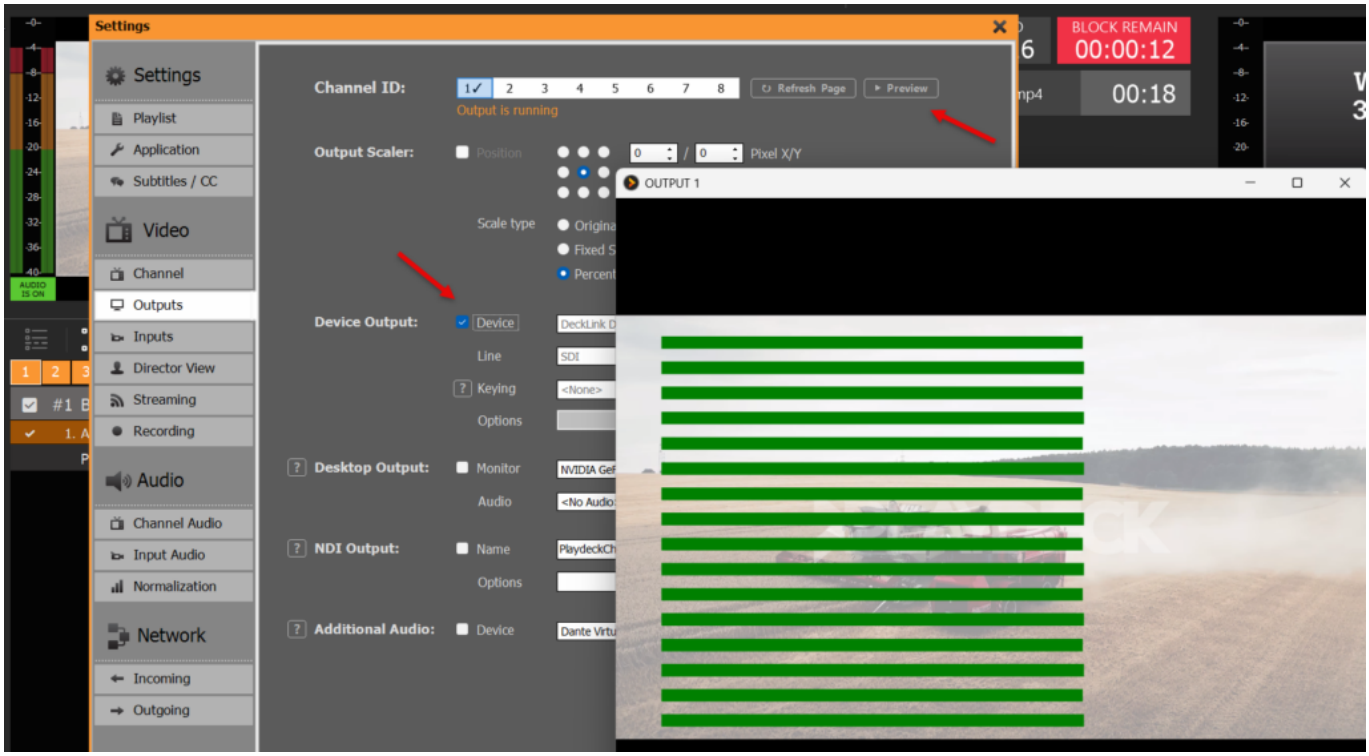
Enable Multichannel Audio by simply setting more than 2 Audio Channel in your Output Channel Settings. In this example we set 16 Audio Channel, as this is the native number of Audio Channel for SDI Output Cards:



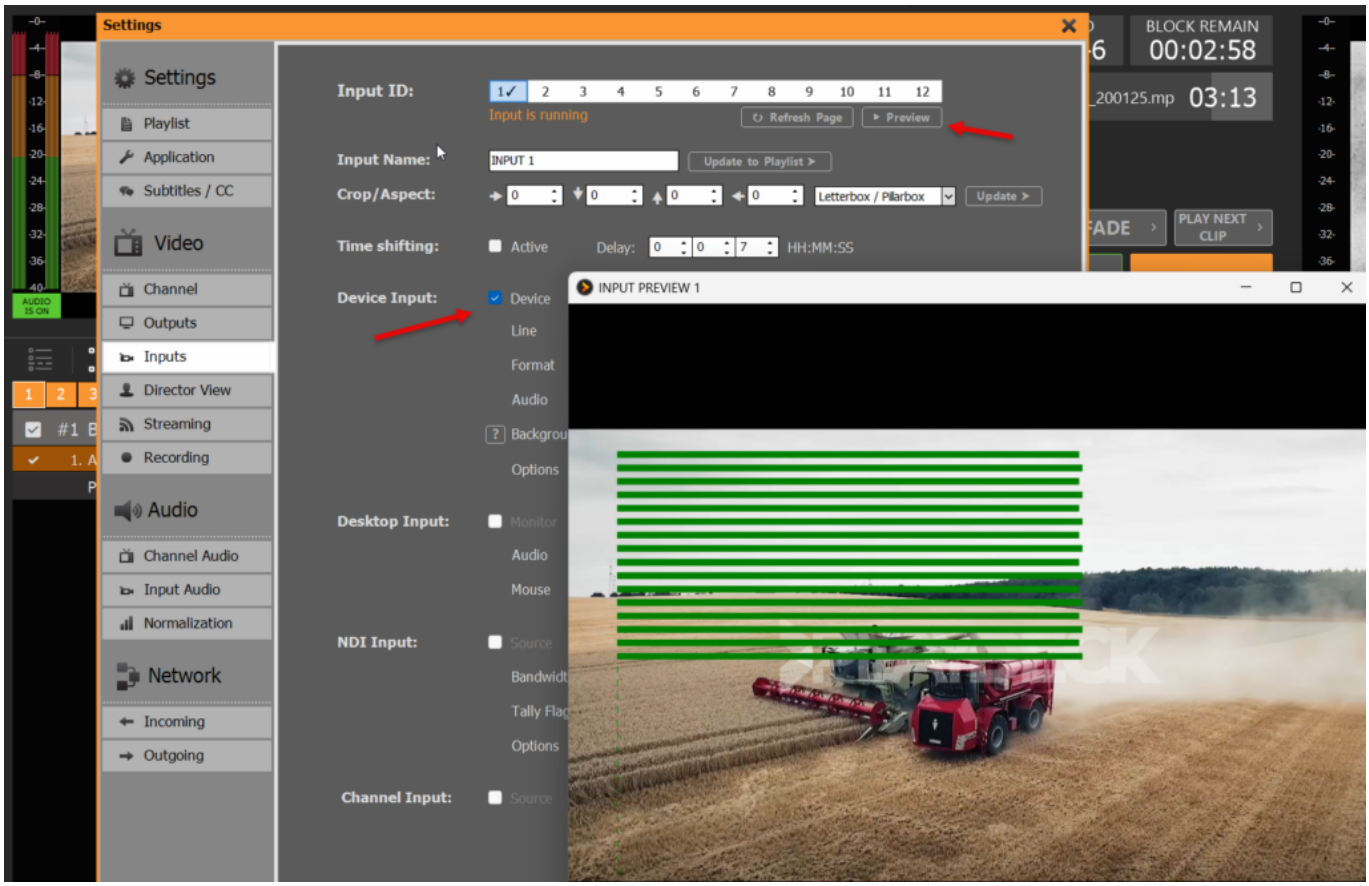
Now you are good to go already. Your VU Meter will switch automatically to 16 Audio Channel:

Multichannel Audio is not also active for all Inputs and Outputs: SDI, NDI, Streams. You can check, if you click PREVIEW after activating your Device.

All Previews have VU Meter as Overlays:



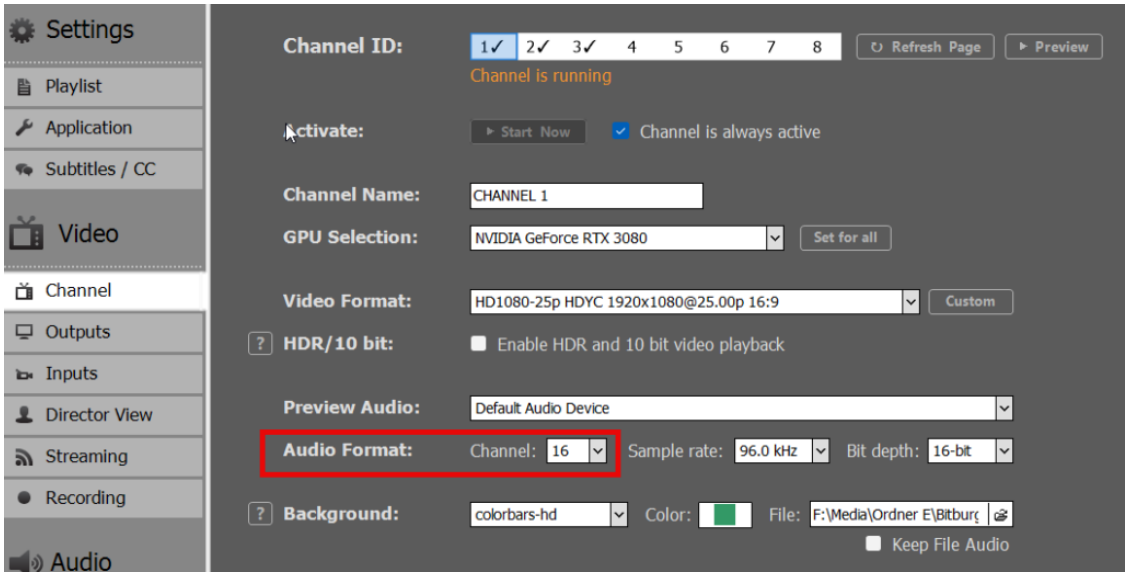
Here is another Example for Input Preview:



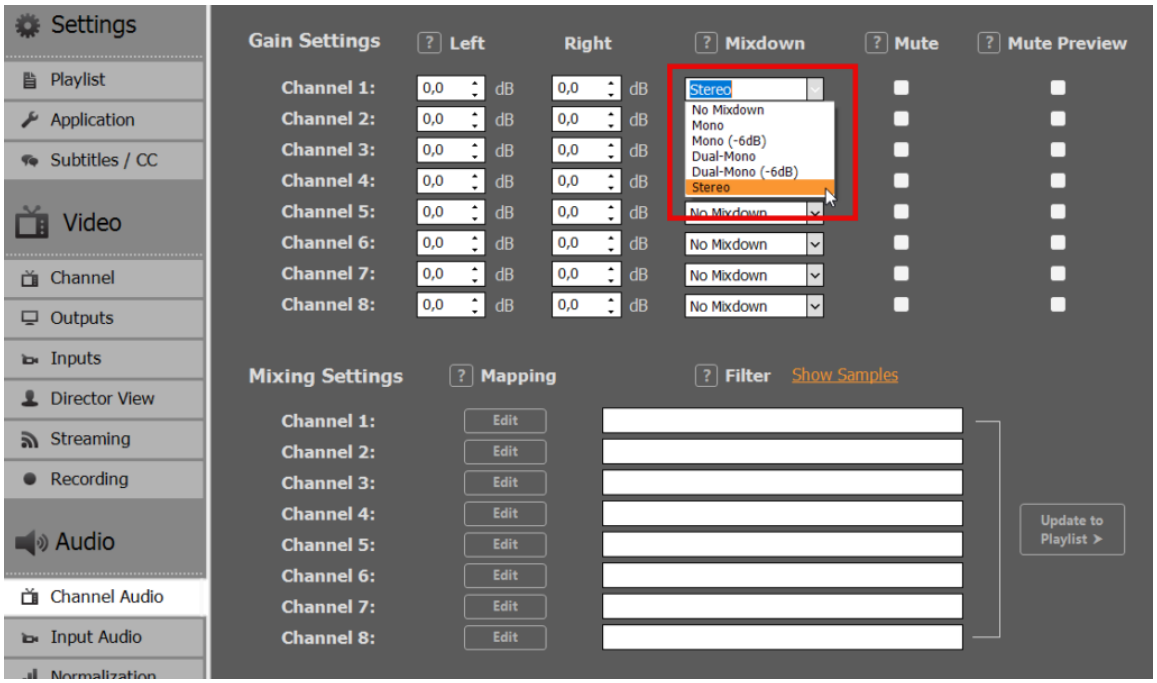
Mix-down

If you have Multichannel-Audio Content (or SDI Inputs) and want to Mix-down

your Audio to MONO/STEREO, you would also need to increase the Audio Channel. In this case we use 16. This is important to tell PLAYDECK to process 16 Audio Channel (from Source eg SDI), otherwise all Audio Channel above 2 would be CUT/SILENT:

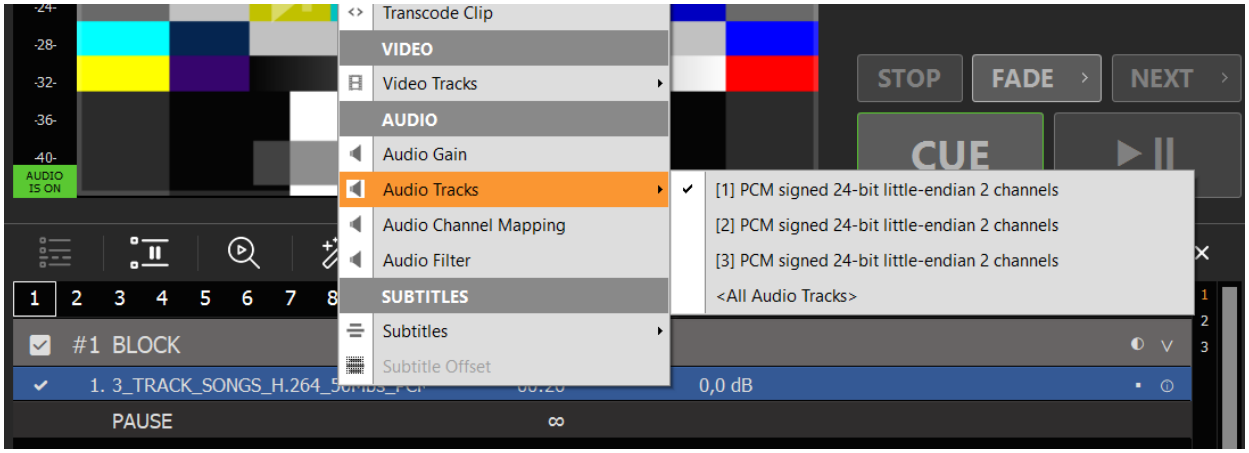


You can now select different ways to Mix-down your Audio. Please note, that there are seperate Settings for CHANNEL (Clips, Input Streams) and INPUTS (Device Input):



Multiple Audio Tracks

If your File has more than one Audio Tracks, you can switch the Audio Tracks by right-clicking the File and selecting AUDIO TRACKS:

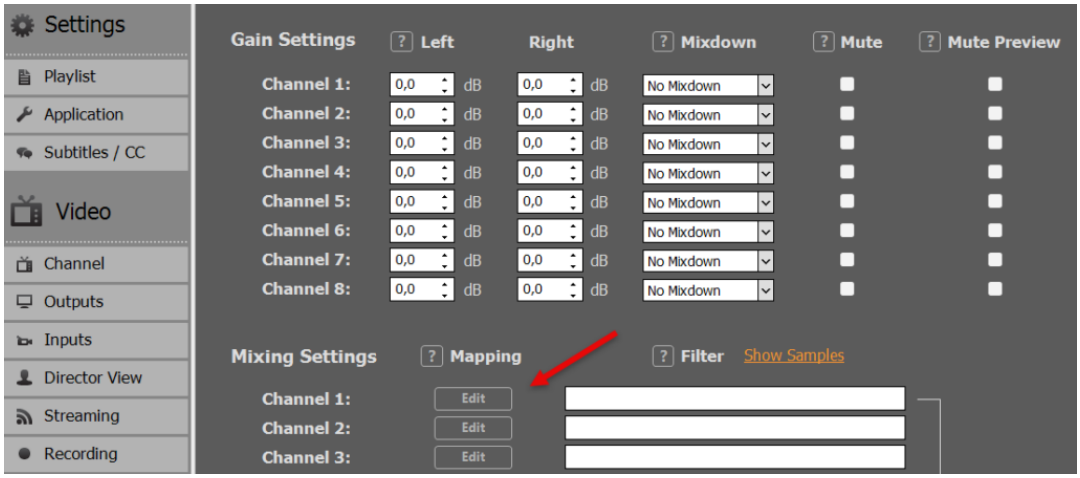


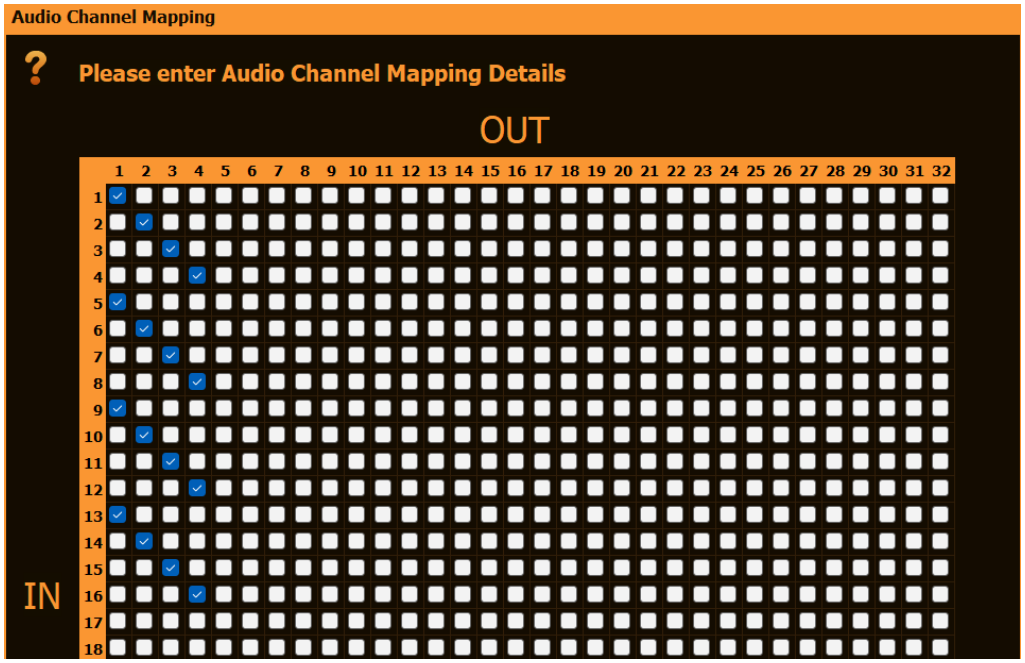
To play ALL Audio Tracks at once, select ALL AUDIO TRACKS. Alle Audio Channel will be concatenated. In our example this would result in 6 Audio Channel for Output. So make sure to set your Channel to 8 Audio Channels, otherwise everything above 2 Channel will not be processed:

Note: Use any of the other Mixing Options to Mix-down to STEREO or similar.

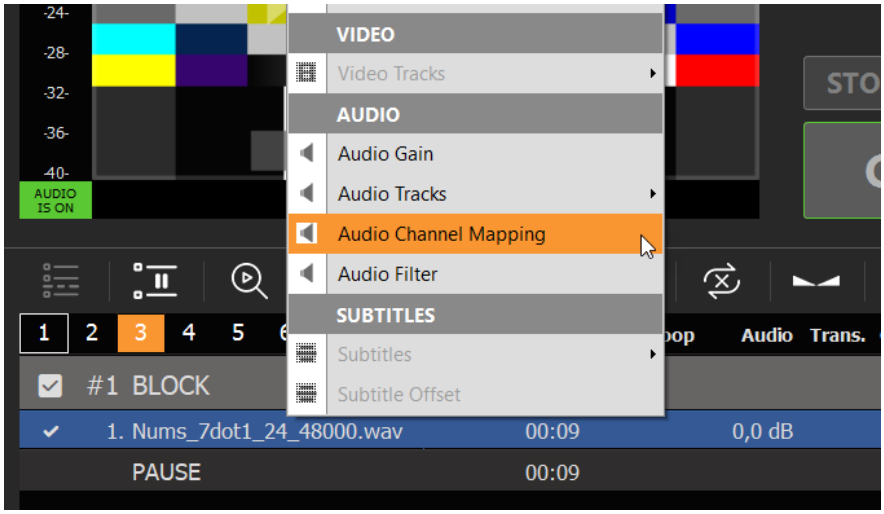
Custom Mixing

For more advanced Mixing, you can click EDIT under MIXING SETTINGS. In this example we Mix-down 16 incoming Audio Channel to 4 outgoing Audio Channel. Please note, that there are seperate Settings for CHANNEL (Clips, Input Streams) and INPUTS (Device Input):





You can refine your Mixing even more by moving to the Content Level: Right-click any Playlist Item and select AUDIO CHANNEL MAPPING:

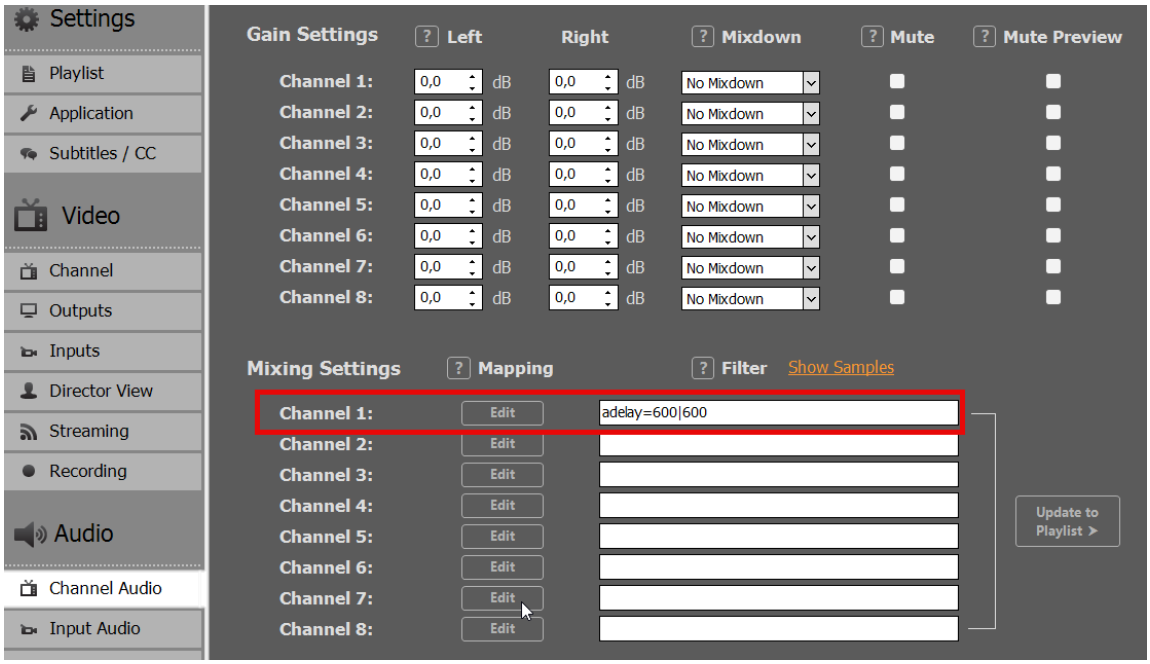


Audio Delay / Filter

Sometimes, in Live Productions, you have to send the Audio delayed compared to the Video, to compensate for the processing Lag of huge LCD Screens (Lipsync). Or add a Limiter/Gate to the Microphone Input Audio.

Simple Audio Delay

This is actually pretty fast to implement in PLAYDECK. You add the ADELAY Audio Filter to the Channel like this. In this example we set 600ms Delay for Audio Channel 1 and 2:



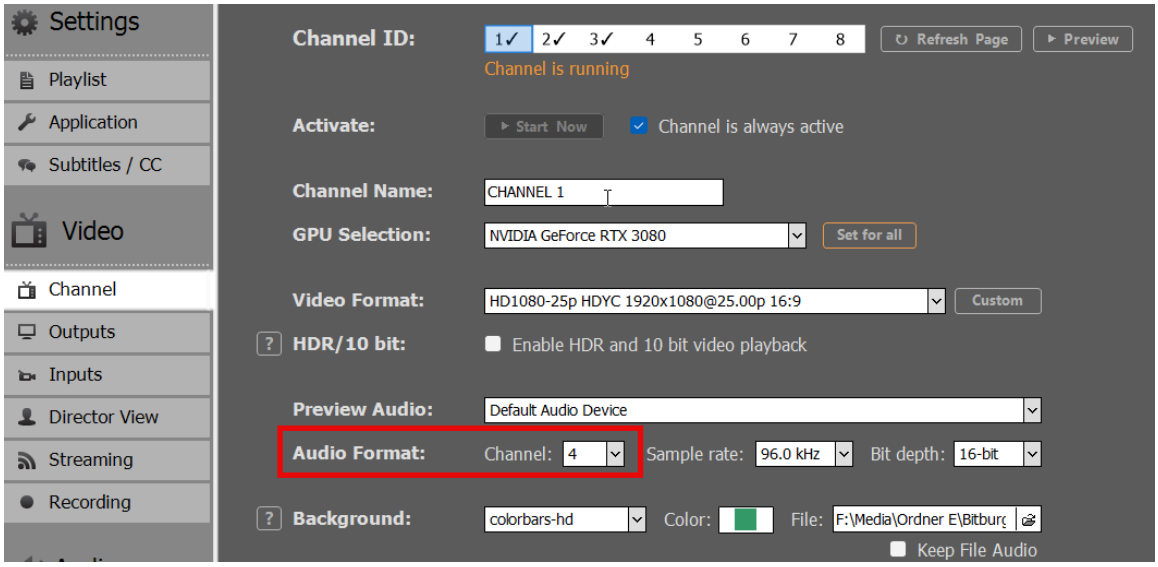
This can also be set to INPUTS to compensate for any incoming Lipsync Issue.

Copy Audio and Delay

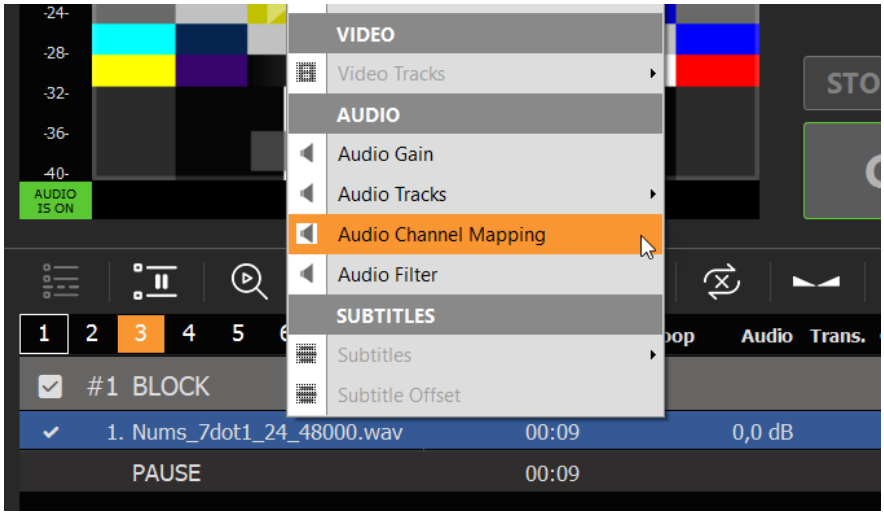
You can also COPY the Audio Channel 1 and 2 to Audio Channel 3 and 4 and delay those. The use case here is, that the Audio Mixer Person has Live Preview Audio.

This needs to be done on the Playlist Level, so clear any Audio Filter in the Settings.

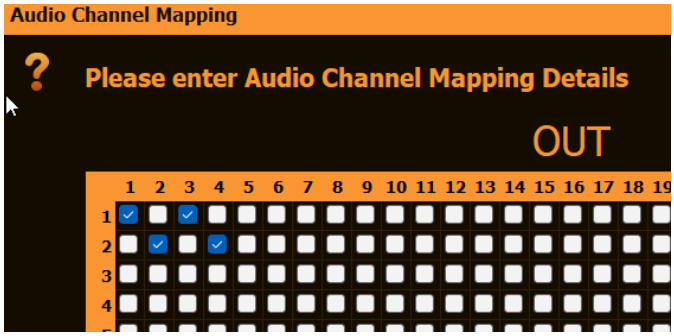
First, make sure to increase the Audio Channel for your Output Channel, otherwise all Audio Channel above 2 will not be processed:



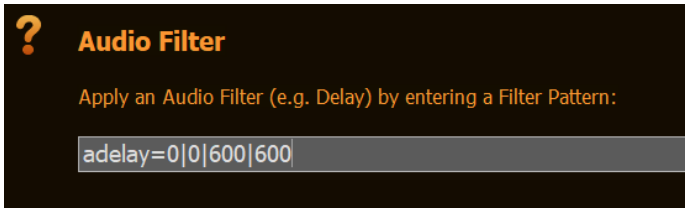
The Right-Click any Playlist Item and select AUDIO CHANNEL MAPPING:



This will copy Audio Channel 1 and 2 to Audio Channel 3 and 4:

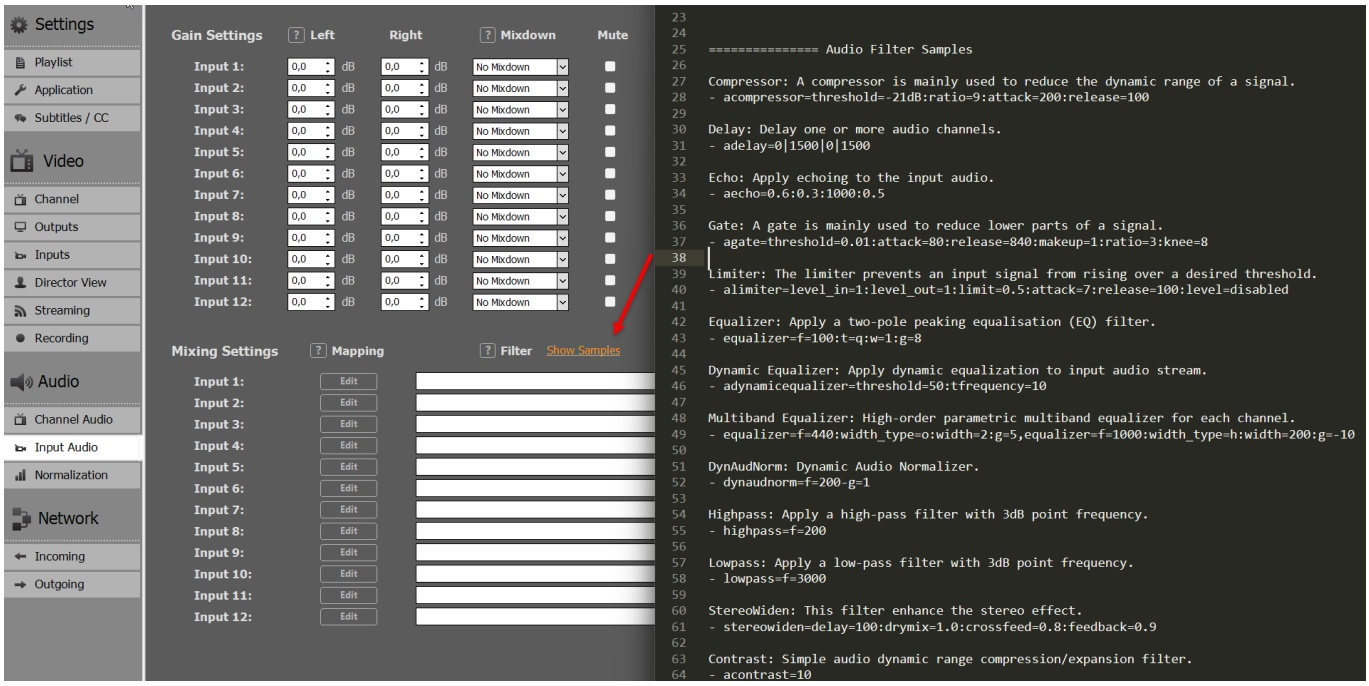


Now right-click the Playlist Item again, select AUDIO FILTER and add the ADELAY Filter, but only for Audio Channel 3 and 4:



More Audio Filter

For a complete List of all Audio Filter, click SHOW SAMPLES. This will open a TEXT FILE with Examples.

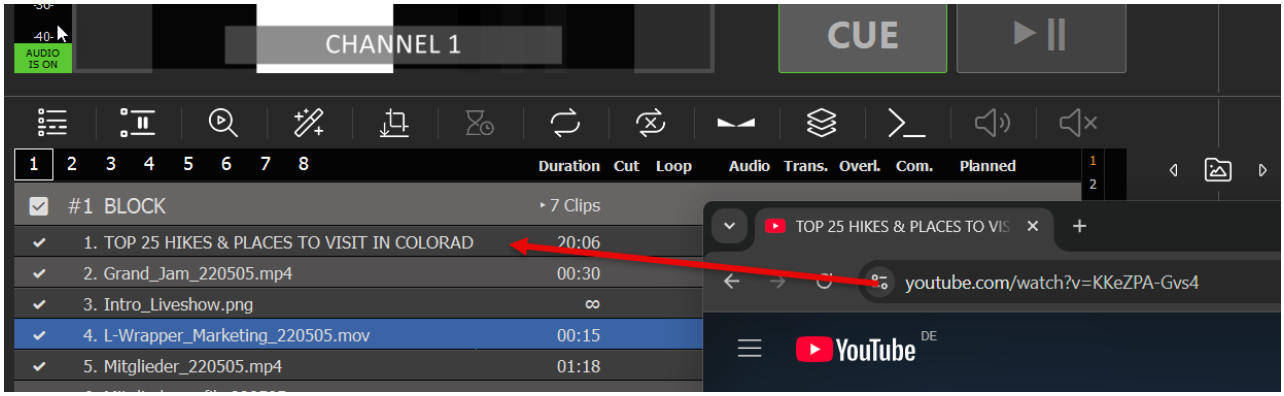


Insert Input Streams into Playlist

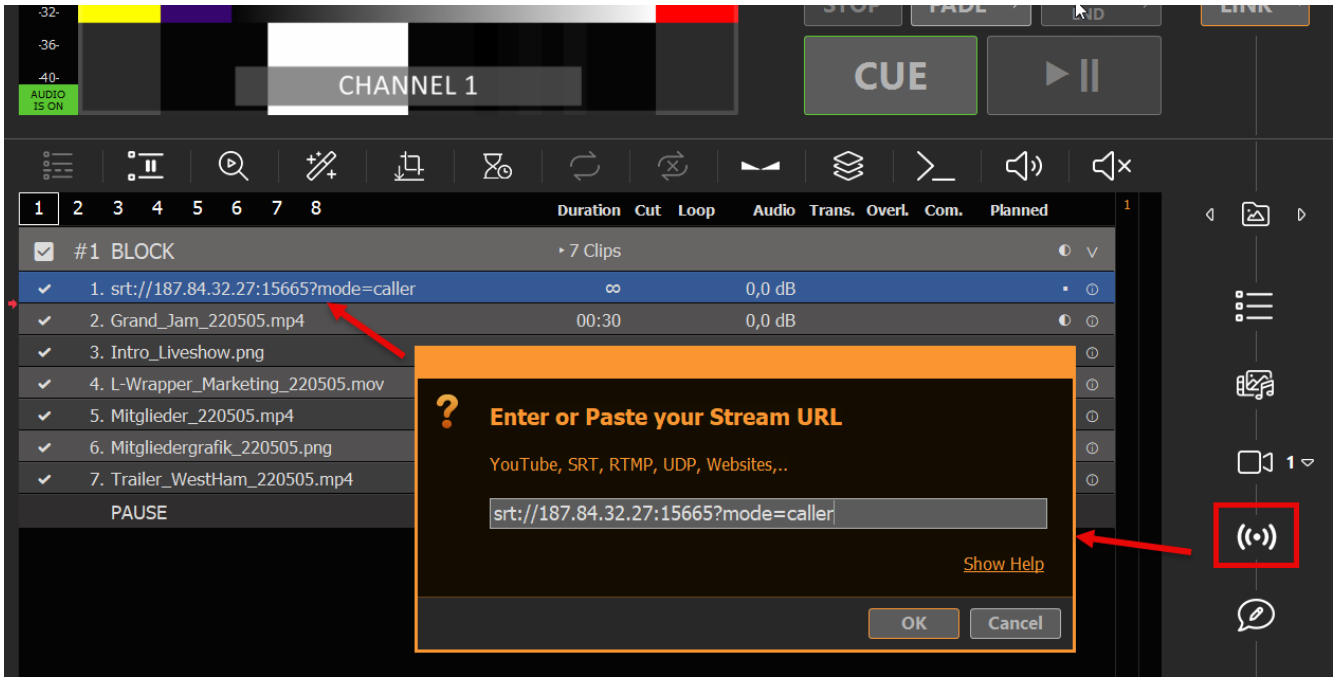
This article will show how to insert Streams into your Playlist.

Insert Stream into PLAYDECK

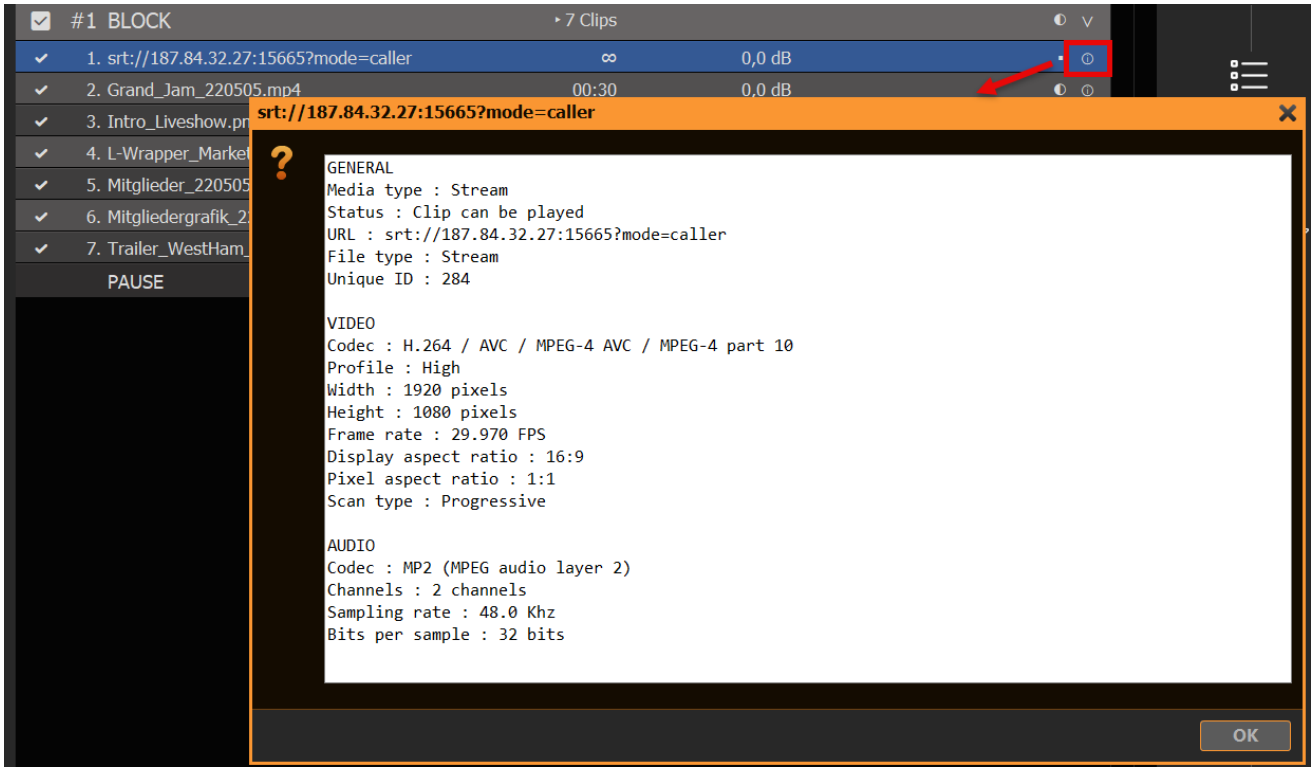
You can directly Drag Drop any YouTube URL from your Browser to PLAYDECK:



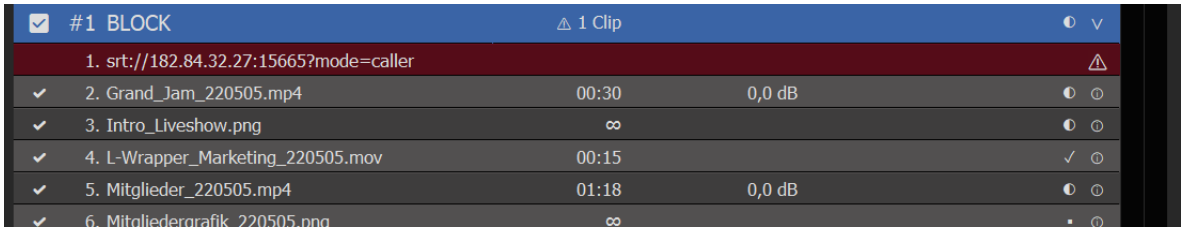
To insert a new Stream into your Playlist, simply Drag Drop the STREAM Icon onto the Playlist. A new Popup will appear, where you can enter the Stream URL:



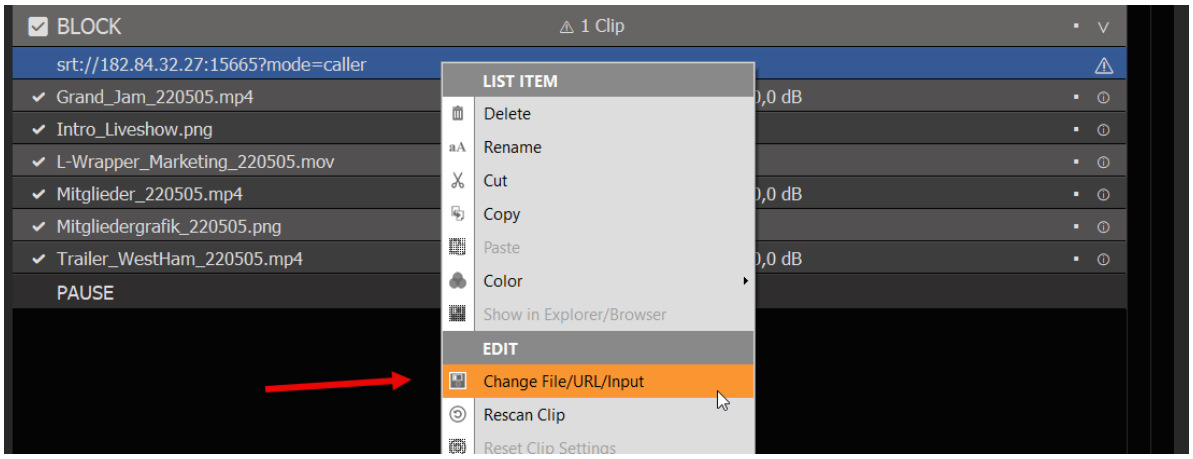
Your Stream will be scanned by PLAYDECK now. If the PLAYDECK could connect to your Stream successfully, the Text will be shown in WHITE. You can now play the Stream and/or can double-click the INFO icon to view more information about the Stream, like Video format and codec:



If the Stream can NOT be connected, it will appear in RED:



If you made a typo, you can quickly re-edit the URL by right-clicking and selecting CHANGE URL:

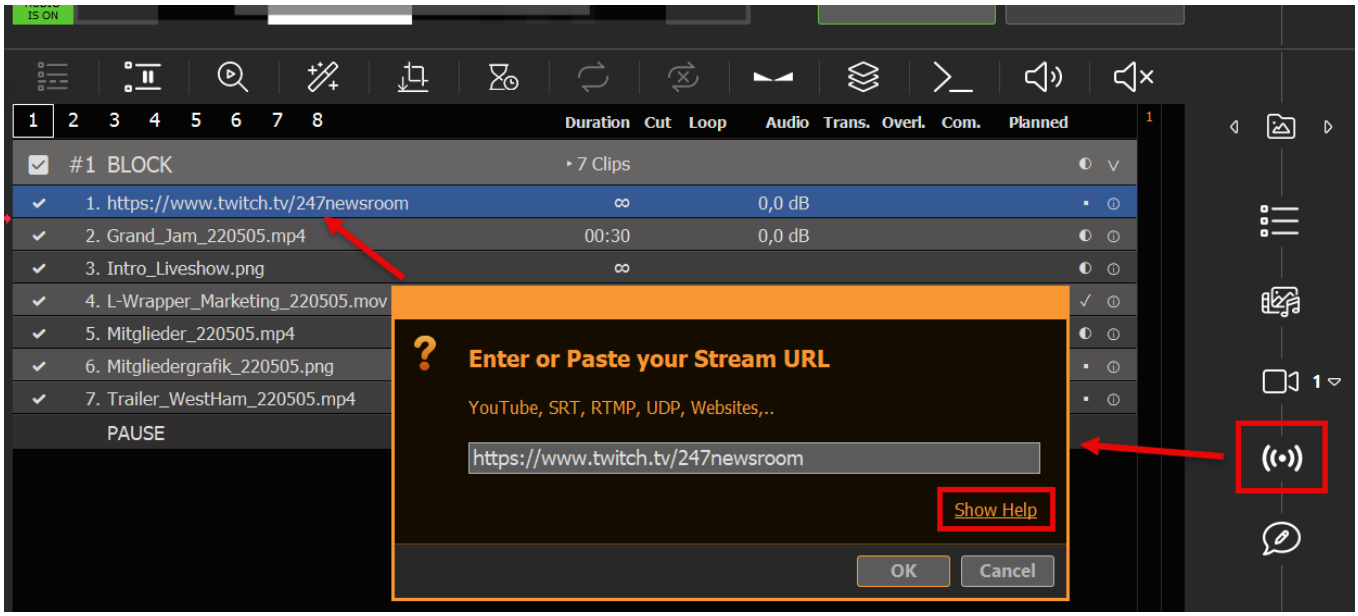


Insert YouTube

You can insert YouTube Links via Drag Drop or in the same manner as insert as Input Stream. See this article on more information about YouTube.

Insert Website

You can insert some Websites per Drag Drop directly into the Playlist or via the STREAM Icon. We use TWITCH in this example. Click SHOW HELP to get more info about what Websites can be used:

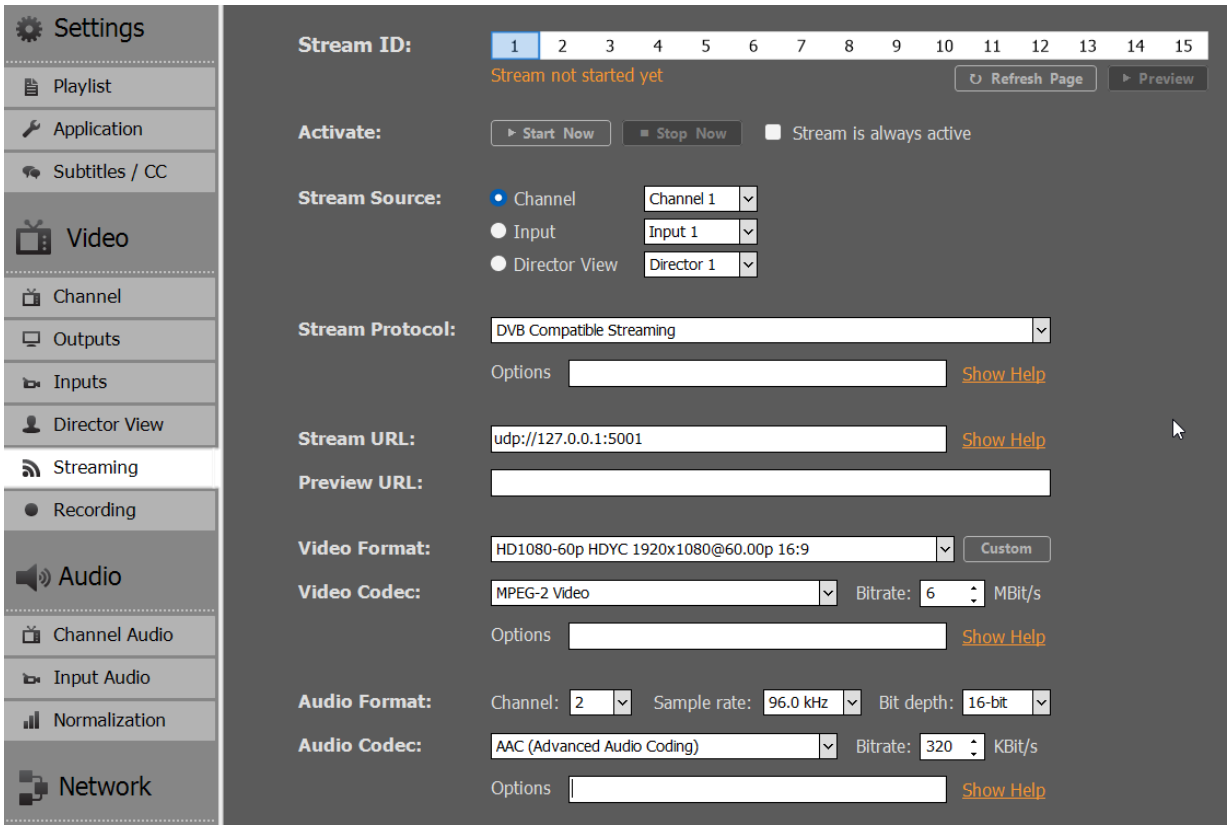


Setup of DVB compatible Streams

This article will show how to properly setup a new DVB compatible Stream.

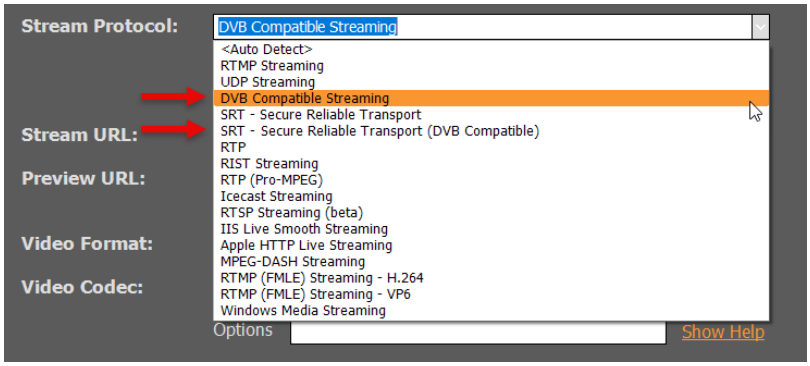
Setup of basic Stream Settings

At first, DVB Streams are created like any other Stream in PLAYDECK, by going into the Settings and entering the appropriate Stream Info. In this case we use a local UDP URL for Testing, a High Framerate of 60 FPS, a medium Video Bitrate of 6 MBit/s, the standardized MPEG-2 Video Codec, a High Quality Audio with AAC Codec and 96 KHz Sample Rate and 320 KBit/s Bitrate:



There are 2 types of DVB Streaming:

- Regular DVB Streaming via UDP (udp://…)
- SRT DVB Streaming via SRT (srt://…)



Additional DVB Settings

DVB Streams rarely work out-of-the-box like other Stream Formats (e.g. RTMP), because the need specific additional Settings, which we walk-thru now. You can also use this official DVB Spec PDF as Reference.

PCR

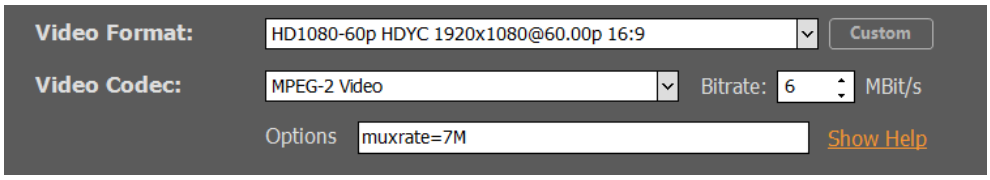
PCR stands for “Program Clock Reference”, which is send by our DVB encoder. It’s kinda a “heartbeat” for the DVB Stream. If your network connection between PLAYDECK and DVB receiver is not reliable and stable, you will easily get PCR errors on your receiver. The PING from PLAYDECK to DVB receiver should never be higher than 20ms, even in peak times (e.g. other traffic).

MUXRATE

You should always set the Muxrate with this Formula:

MUXRATE = (VIDEO BITRATE + AUDIO BITRATE) * 1.25

So in our Sample above with have (rounded): 8 Mbit/s. We set the MUXRATE to PLAYDECK via the OPTIONS field of our Video Codec:



As a result, the resulting Bitstream will be of the MUXRATE value and non-video and non-audio parts are filled with null-packets.

CLOSED CAPTIONS + SCTE-35

Both dont need to be activated specifically, as they are automatically enabled by PLAYDECK.

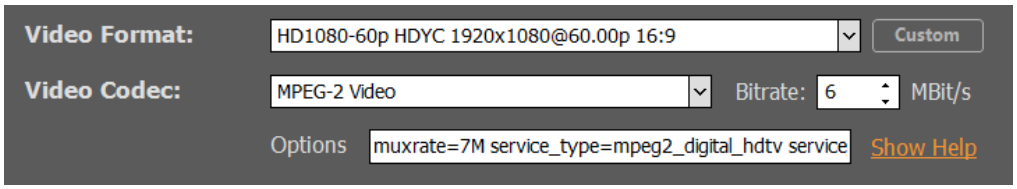
INFORMATION TAGS

There are several informative fields, which are unique to DVB:

- service_name (any text value)
- service_provider (any text value)
- *service_id* (any number – default is 1)
- *service_type* (possible values: digital_tv, mpeg2_digital_hdtv, advanced_codec_digital_sdtv, advanced_codec_digital_hdtv, hevc_digital_hdtv – default is digital_tv)
- *transport_stream_id* (any number – default is 1)
- original_network_id (any number – default is 1)
- *pmt_start_pid* (numbers between 16 to 7936 – default is 129)
- *start_pid* (numbers between 256 to 3840 – default is 1024)
- *pes_payload_size* (any number – default is 2930 bytes)
- *mpegts_flags* (possible values: resend_headers, pat_pmt_at_frames, latm)

- *start_timecode* (possible values: auto, disabled, local_time OR custom text like “10:00:00:00” – default is auto)
- *tables_version* (any number – default is 0)

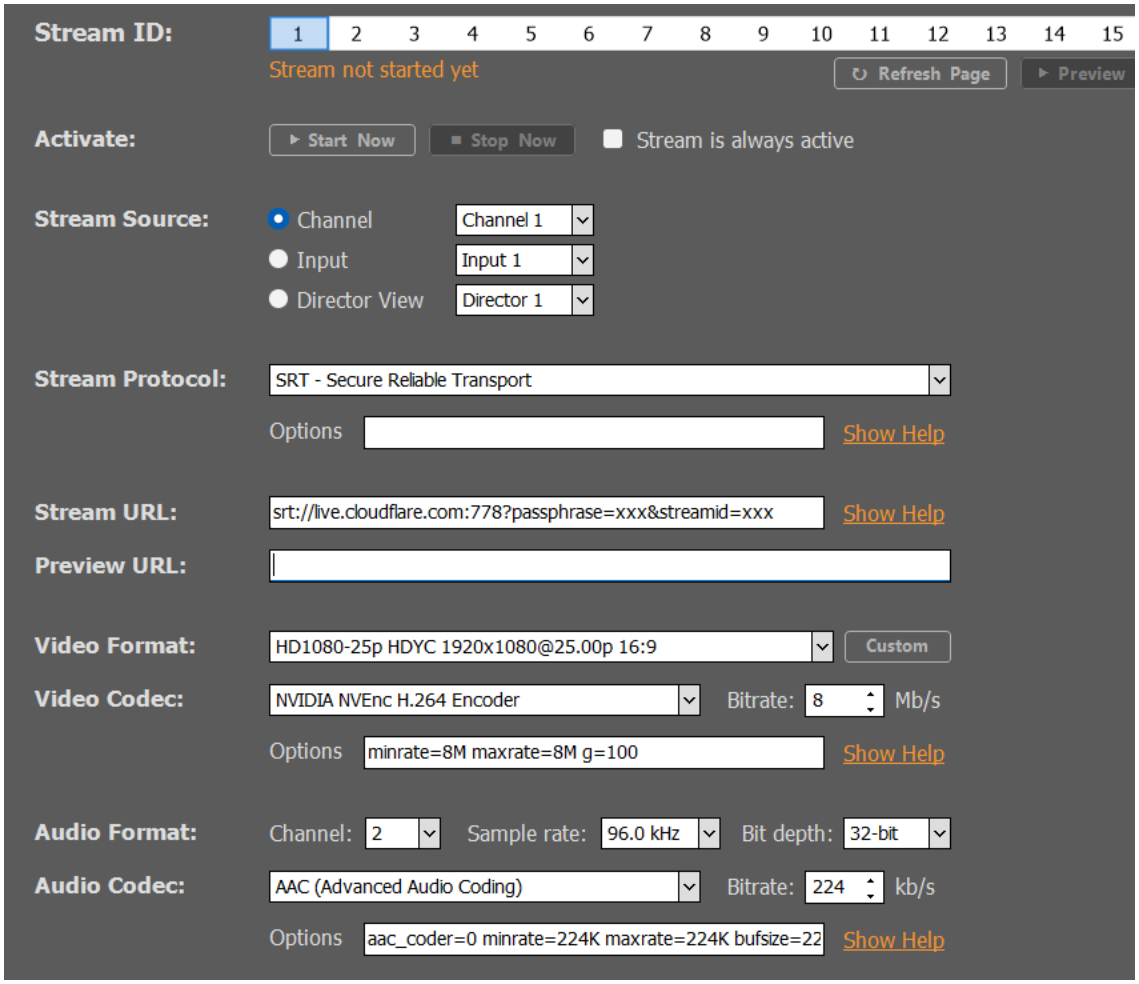
These values are also entered into the OPTIONS field of our Video Codec. You simply use a SPACE between options. Add as many options as you like/need:



High-end SRT Streams to Cloudflare CDN

This article will give pointers on how to improve your Streams even more. In this example we use Cloudflare, but this can be any provider.

These are the optimized Settings as overview. We will explain them in detail:



Video Format

It is recommended to select a specific Video Format instead of using AUTO, just to avoid any problems with automatic detection.

Video Codec

We know that Cloudflare support H.264 Encoding, so we can use our GPU. If you dont have NVIDIA or your GPU is maxed with other Tasks already, select “Intel QuickSync SW H.264 Encoder”, which will try to use your Onboard Intel-GPU and falls back to CPU otherwise.

We then pick 8 Mb/s as Bitrate, which is a Cloudflare CDN requirement.

With the Options “minrate=8M maxrate=8M” we basically force CBR (Constant Bit Rate) to our Stream. Click this Link on more Info about CBR with H.264.

We also add the Option “g=100”, which sets the Key Frame Interval, also know as “GOP (Group of Picture) length” or “IDR period”. The rule-of-thumb here is to set the GOP twice as high as your framerate, which would be 50 in this case. We use 100 here to compress even more, resulting in less used bandwidth, which in turn makes your Stream more stable. A higher value also reduces buffering, but not all provider can process high values. Reduce your GOP, if you Content has alot of Action and your need more key frames. The default GOP is 15, which is a failsafe to ensure, that there are no artifacts in the picture.

Audio Format

We use 96 kHz Sample-rate and 32-bit as Bit-depth, which will make absolutely sure, that all possible audio sources are processed at high quality. Since the Audio Stream has a very low impact on the overall bandwidth, it would be unwise to go low.

Audio Codec

There is not much choice here for the Encoder, as AAC delivers high quality low compression audio. Since we picked a higher Sample-rate and Bit-depth, we need to provide a bit more Bitrate here, so we set this to 224 kb/s.

As Options, first we set our AAC Encoder to the “TLS” mode by entering “aac_coder=0”. The Options here are:
0 – Two loop searching (TLS) method
1 – Average noise to mask ratio (ANMR)
2 – Fast constant quantizer method
TLS will have the highest quality with most audio contents. The default here is “2 (fast)”, which is also good, but can lead to crushing sounds at lower bitrates.

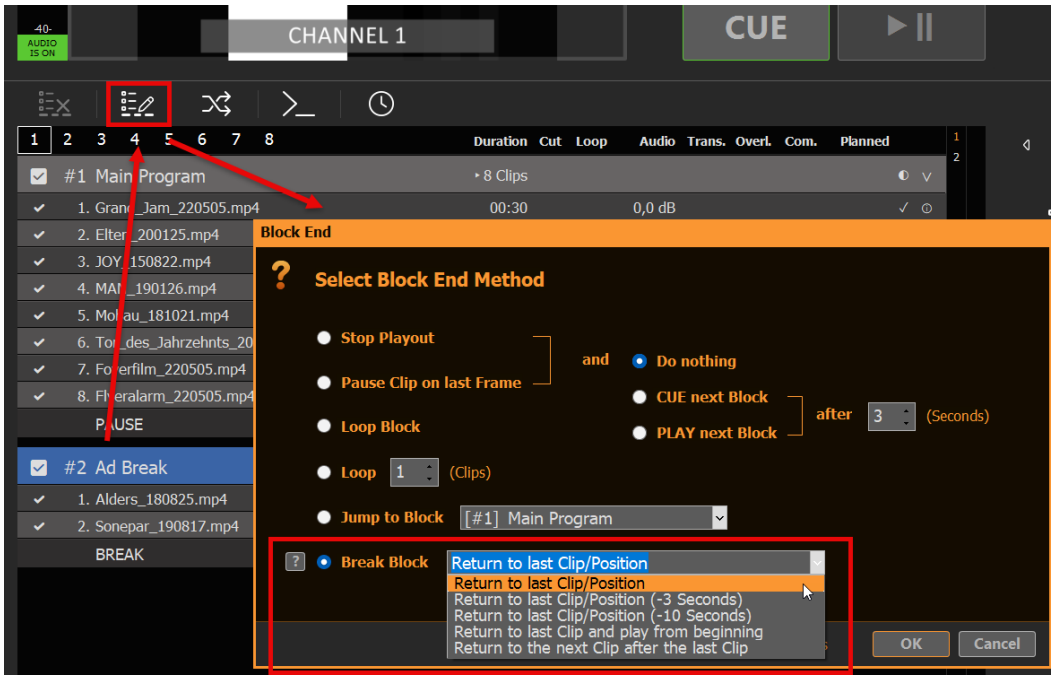
Next we also force CBR for our Audio Stream by setting “aac_coder=0 minrate=224K maxrate=224K bufsize=224K”. Click this Link for more info on CBR for AAC.

Ad Breaks to interrupt Playout

This article will show how to use a BREAK BLOCK to create Ad Breaks, that interrupt your regular Playout, and wil return after.

Using Break Blocks as Ad Breaks

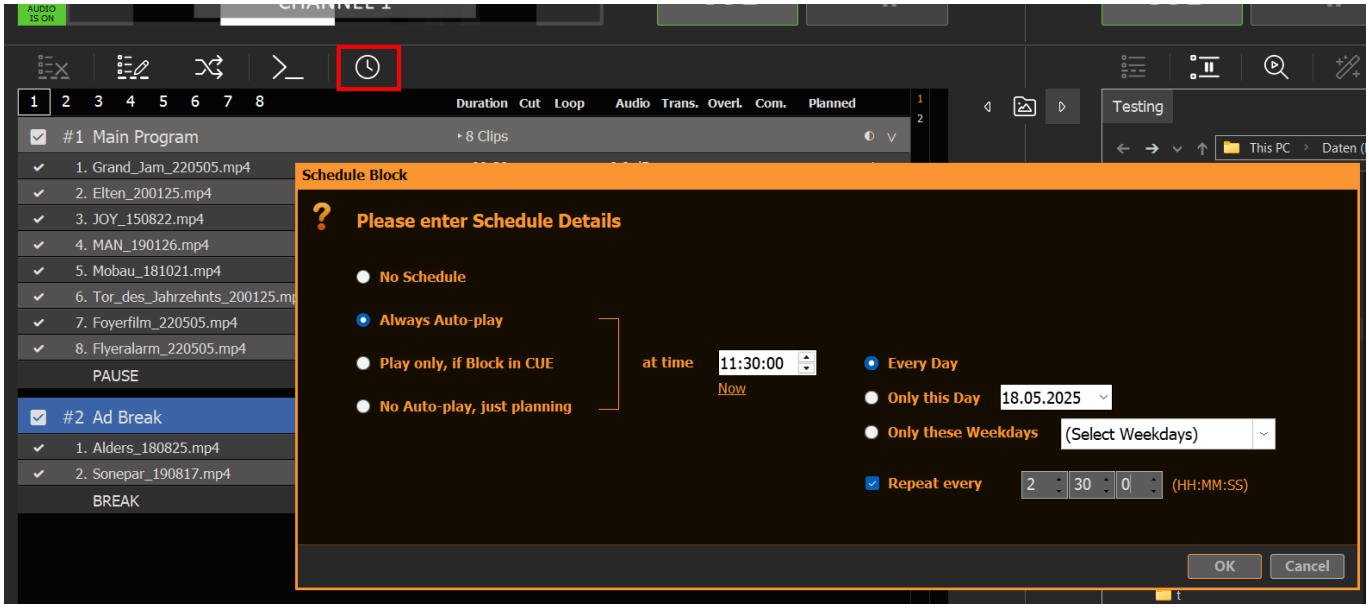
This is fairly simple: You create a new Content Block in your Playlist with one difference: The Block ends as BREAK BLOCK (instead of STOP or PAUSE). For this we select the Block and click on the BLOCK END Icon (or Double-Click the Block end). Then we select BREAK BLOCK and choose RETURN TO LAST POSITION (or any other):



This BREAK BLOCK behaves like this: Once it reaches its end, it will return to the clip and position (or other), which was playing BEFORE entering the BREAK BLOCK from OUTSIDE. Meaning: You can jump from anywhere to anywhere in the BREAK BLOCK, and it will remember where to return to.

Schedule Break Block

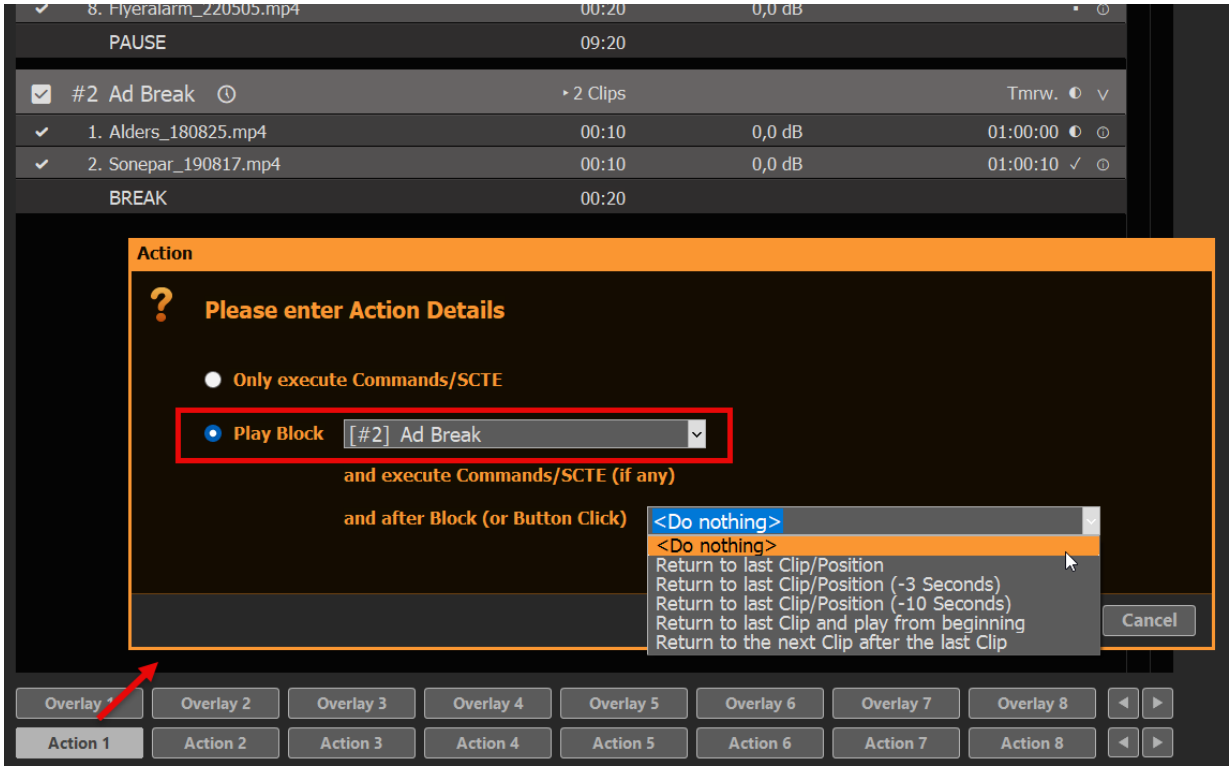
You can schedule the Break Block like an other Block Type by selecting the Block and clicking the SCHEDULE Icon:



In this example we start our Break Interruption every day at 11:00 o’clock and will repeat this every 2 and half hours for the remaining of the day. There will be no Ad Breaks between 0:00 and 11:30.

Using Action Buttons for Ad Breaks

Instead of scheduled Break Blocks, you can also use Action Buttons. For this simply click and empty Action Buton and select to play the related Ad Block:



You get the same selection for “returning” as for Block Ends. If you select “<Do nothing>”, the Block End will be used, like in our example above. If you select any other Option, the Return method of the Action will be used over the Block End method. This gives new options:

- You could use both return methods, depending if the Break Block starts via scheduling or via Action Button
- If started via Action Button, you can leave the Break Block EARLY and still return

The second option is perfect for looping Break Blocks, as your return to the previous playout, if you click the Action Button a second time.

Attach Overlays to the Break Block

Oftentimes you want Overlays to start automatically together with your Break Block Content. We therefore recommend the following articles:

- Automate Overlay Lower Third with Clips
- Create L-Band Ads via second Channel
- External Overlays with ClassX (and other)

Attach SCTE Marker to the Break Block

If you want to signal your Streaming Server or Broadcasting System to insert Ads for the duration of your Break Block, we recommend to following articles:

- Using SCTE for Inputs and Outputs
- Sending SCTE-35 to Stream Server

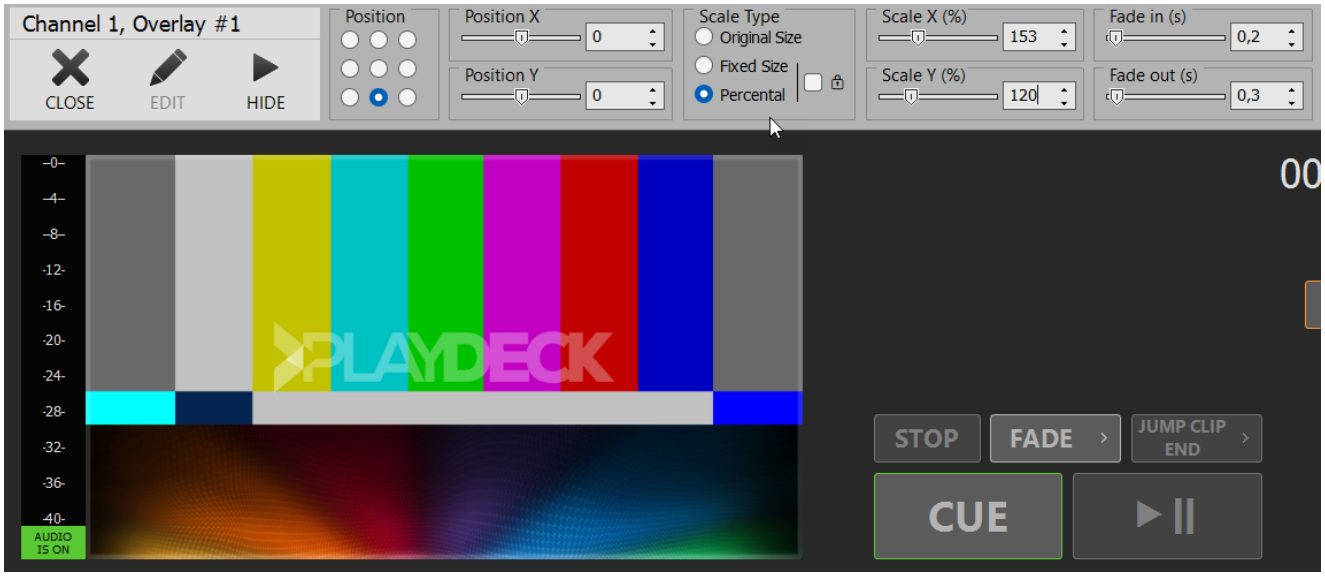
Automate Overlay Lower Third with Clips

This article will show how to use how to create Overlays Groups and bind them to Clips or schedule them.

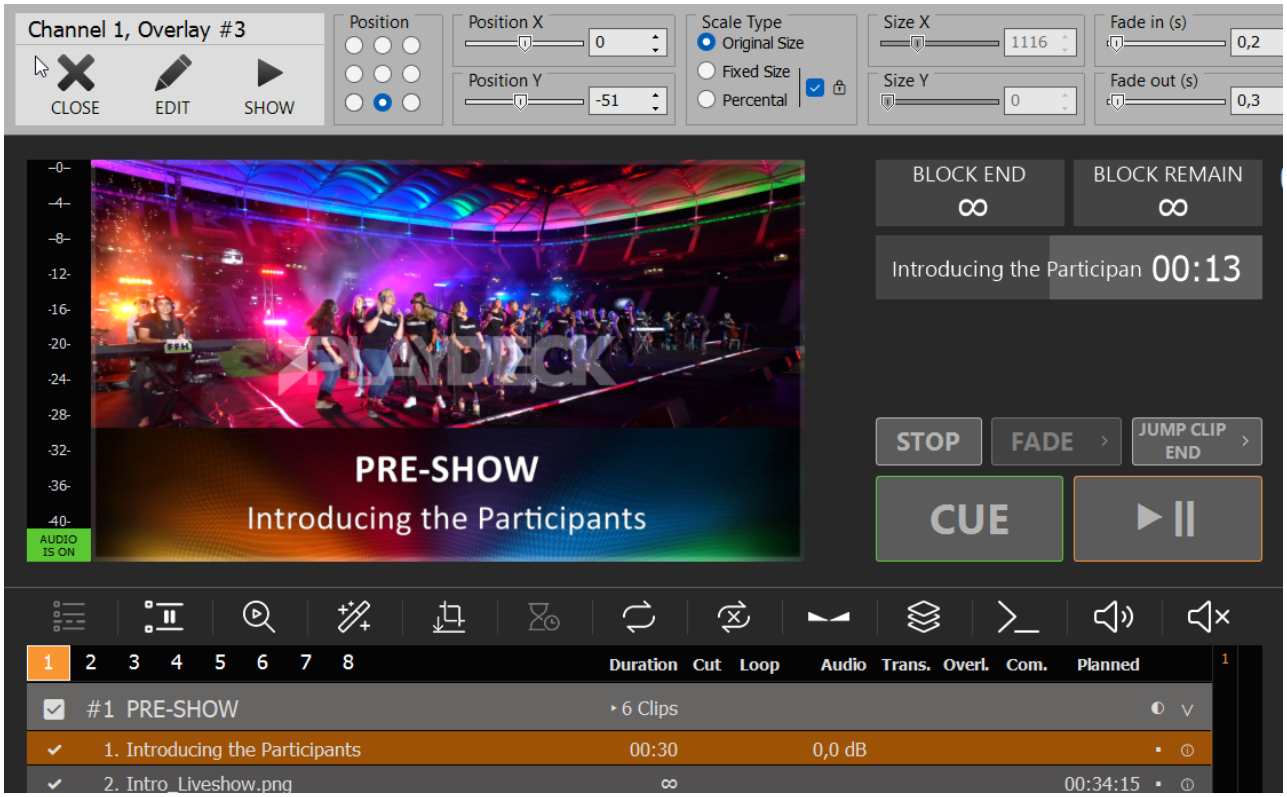
Create some Overlays

Lets create our own custom “lower third” composed of a background GFX and Text and group them together.

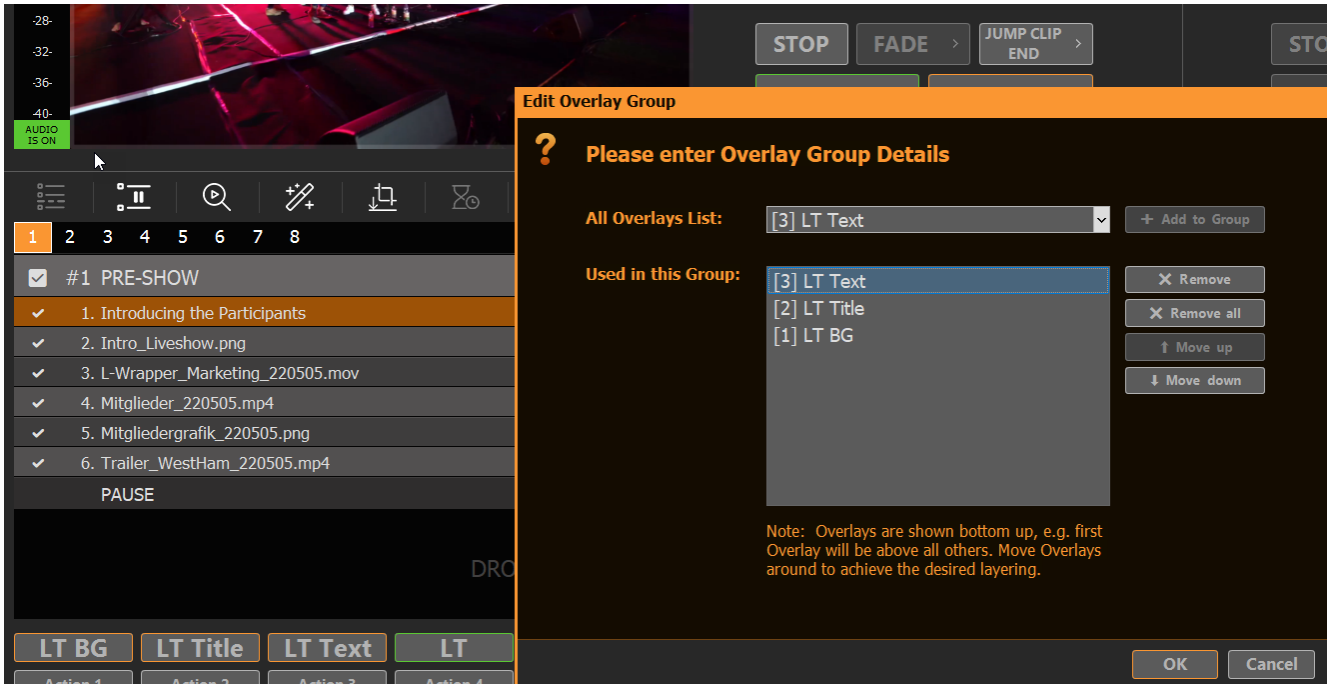
Click on an empty Overlay Button to open the Editor, then add some Background GFX. Close the Editor, right-click the Overlay Button and position your Background to the bottom area. We also right-click in the video preview and disable “Channel Name”:



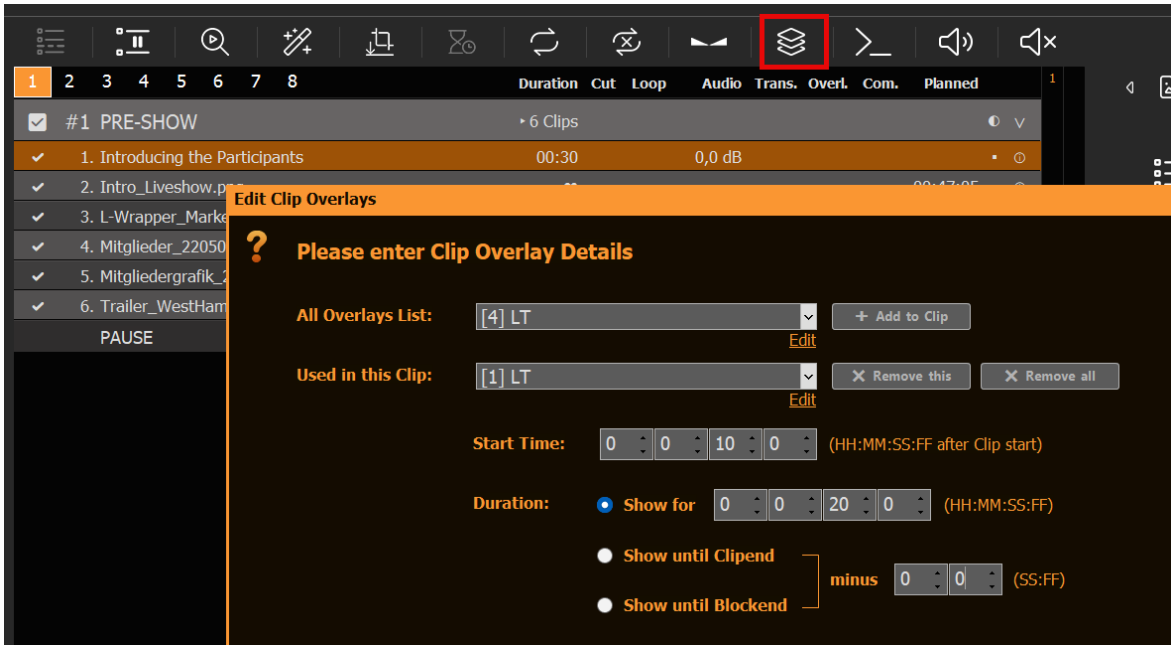
We add another Text Overlay and use {blockname} as text placeholder for our title text. Then add another Text Overlay and use {clipname} as text placeholder. Then some positioning and it looks like this:



Now lets group then together: Right-click an empty Overlay Button and select “Create Overlay Group”. Then add our 3 previous Overlays and order them bottom-up:

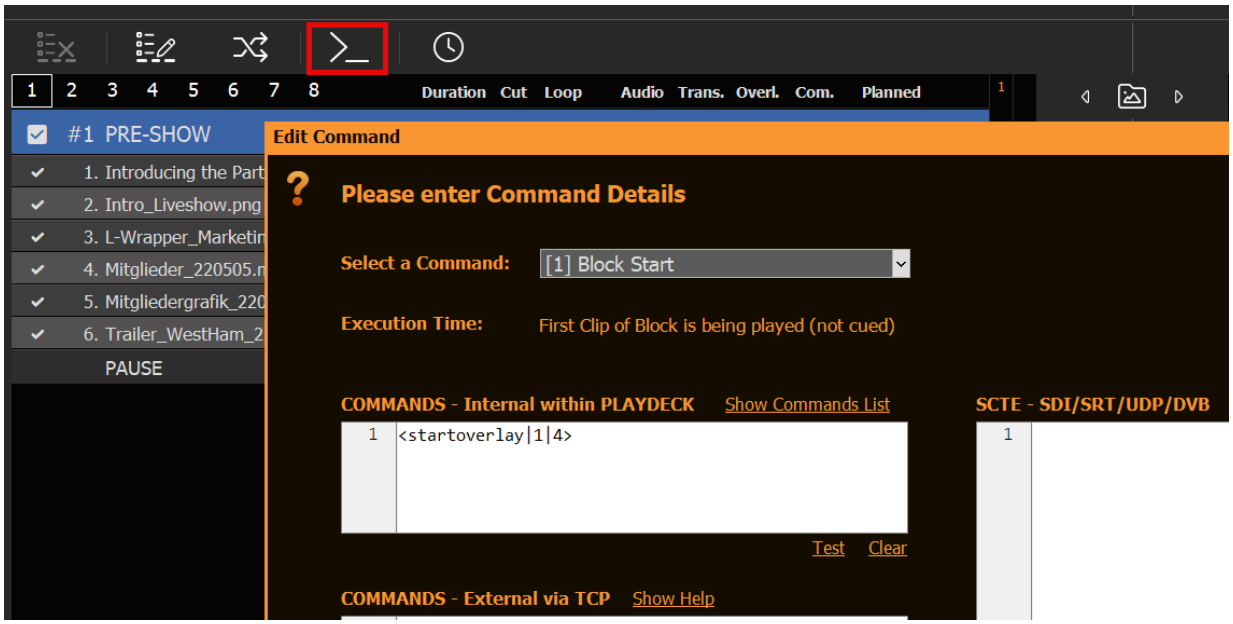


We now have a functioning Overlay Button for the Lower Third with dynamic Text Content. Not lets automate this Group with the Clip. Select the related Clip and click on the Overlays Icon, then add our Overlay Group to that Clip, starting 5 Seconds into the Clip and showing for 10 Seconds:



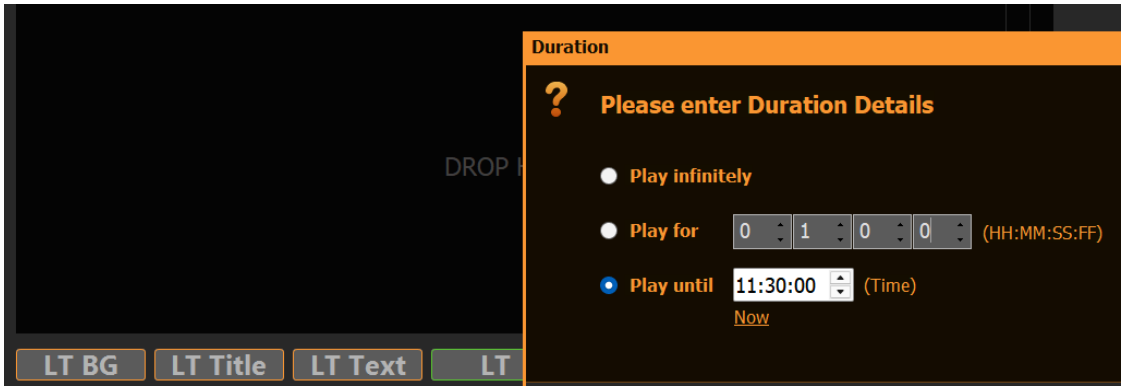
Automate Overlay with Blocks

Since there is no Overlay Icon with Blocks, you use a Command to start/stop the Overlay:



Schedule Overlay

You can show/hide Overlays indendent of Content by entering start time and duration. Right-click on the Overlay and click “Autostart” and later “Change Duration”:

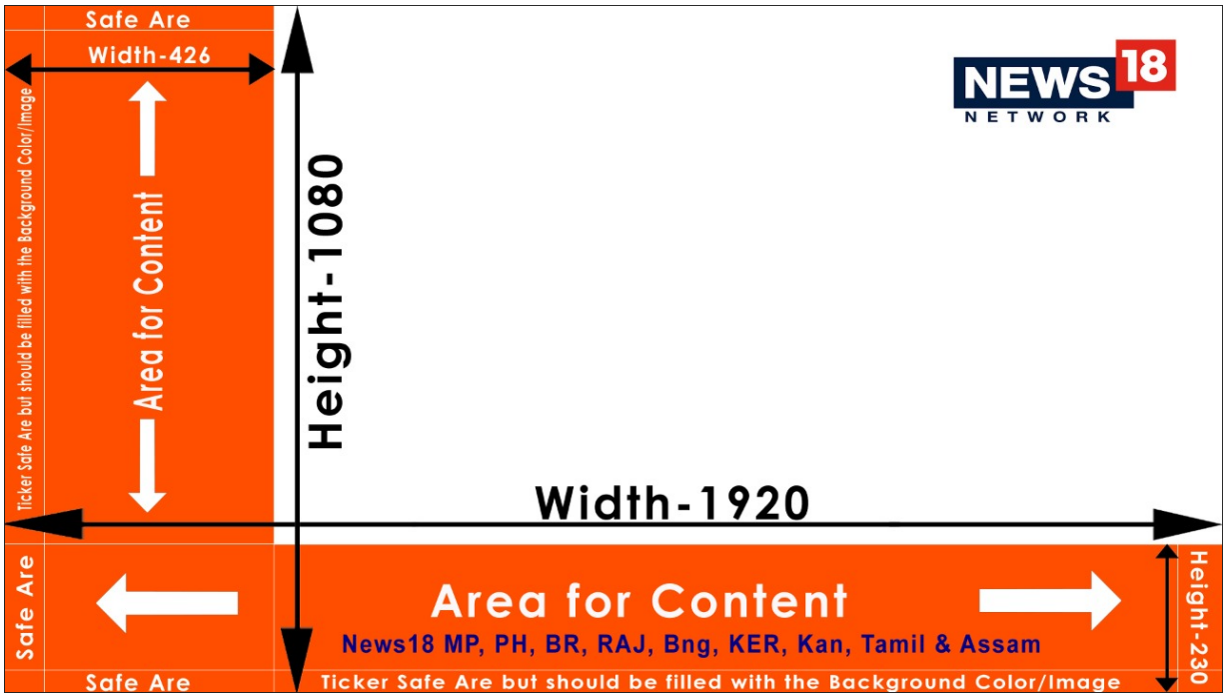

DROP
LT BG
LT Title
LT Text
LT
Duration
Please enter Duration Details
Play infinitely
Play for
0
1
0
0
(HH:MM:SS:FF)
Play until
11:30:00
(Time)
Now

Create L-Band Ads via second Channel

This article will show how to use how to utilize the second Channel to create L-Band avertisements.

L-Band Specifications

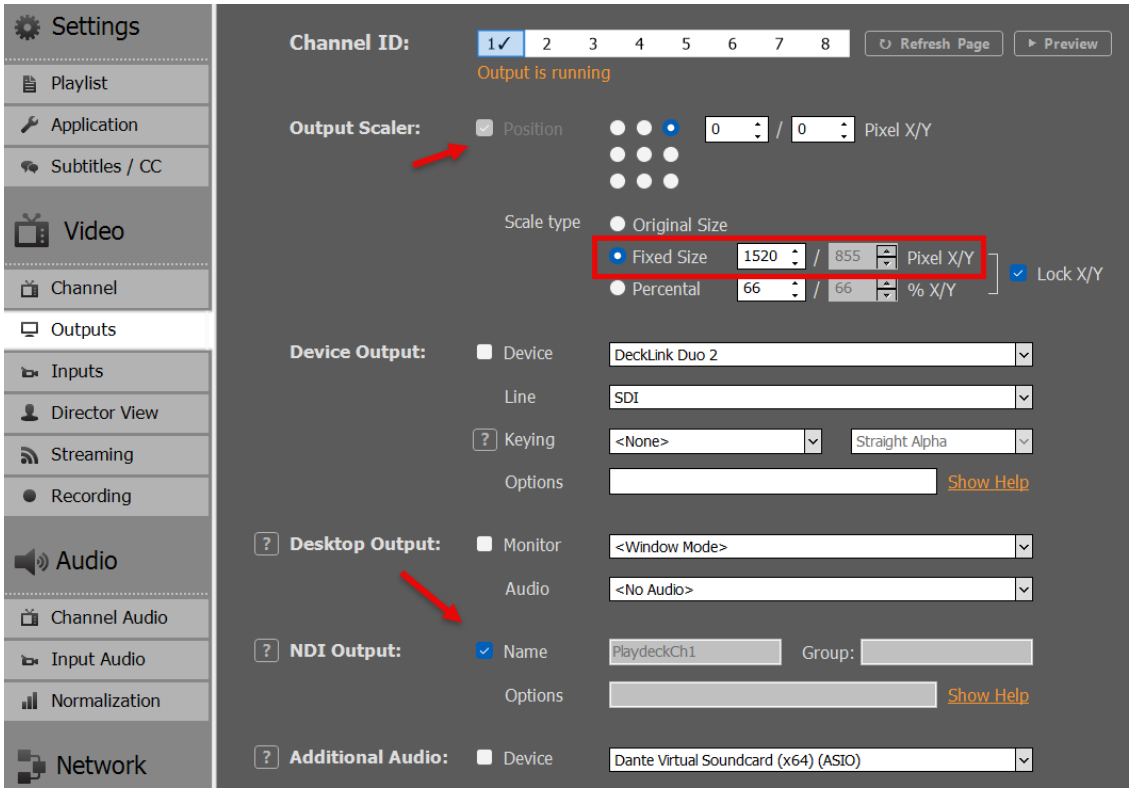
Every Station has their own specs on the size, so we just borrow this sample from News18:



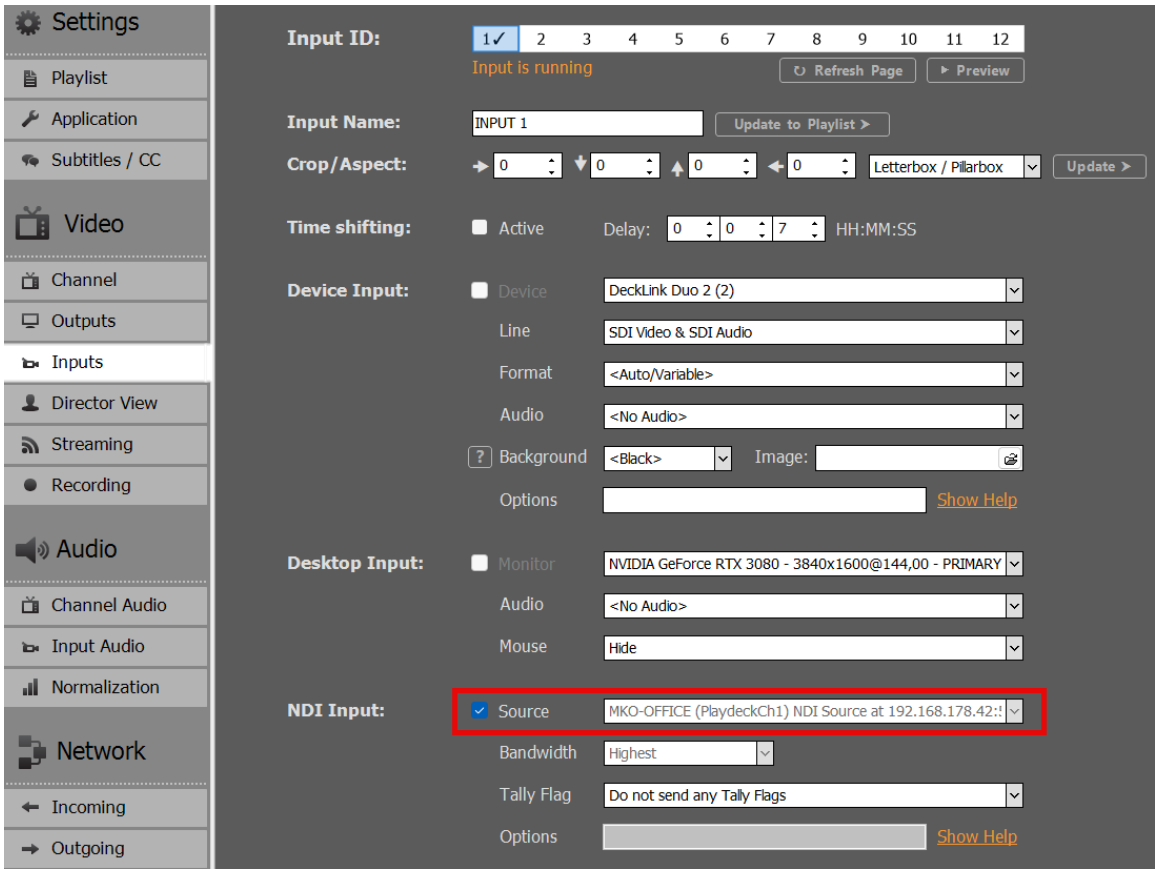
The principal is always the same: We “shrink” our main content proportionally to make room for Ads in the remaining area. After shrinking, we leave some overlap to not risk black background. Our final shrink-size is: 1520×855 Pixel.

Scale Channel 1 and send to Channel 2

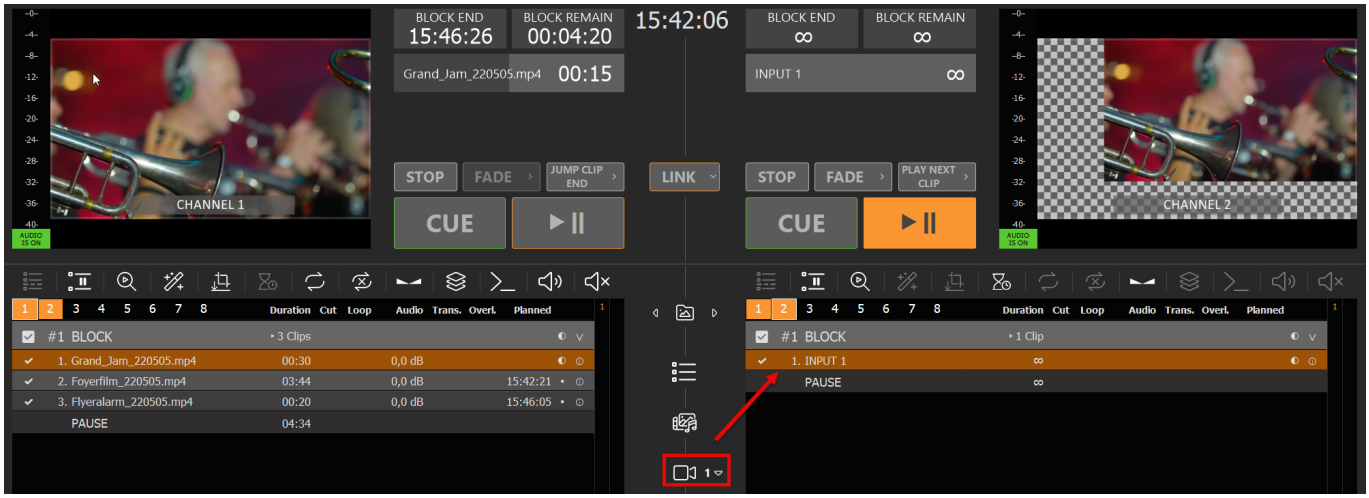
We assume with have our Main Video Content on Channel 1. So we enable the Output Scaler in the Settings to our shrinked size of 1520×855 Pixel. Then we send our scaled Content as NDI signal:



We now loop our NDI signal to Input 1:

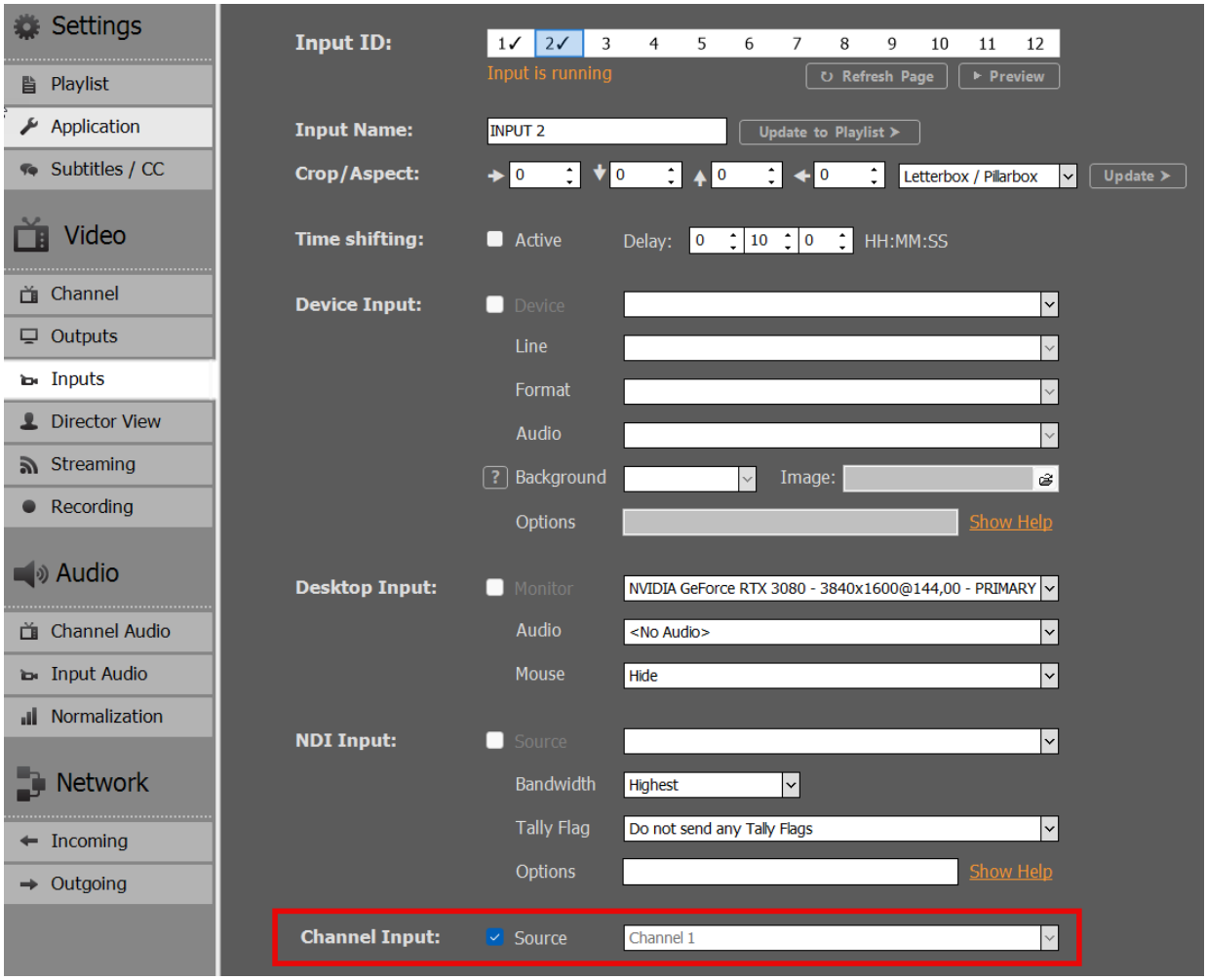


We then add Input 1 to the Playlist of Channel 2, by Drag Drop of the Input Icon to the Playlist. We now have the Video Content in L-Band size:

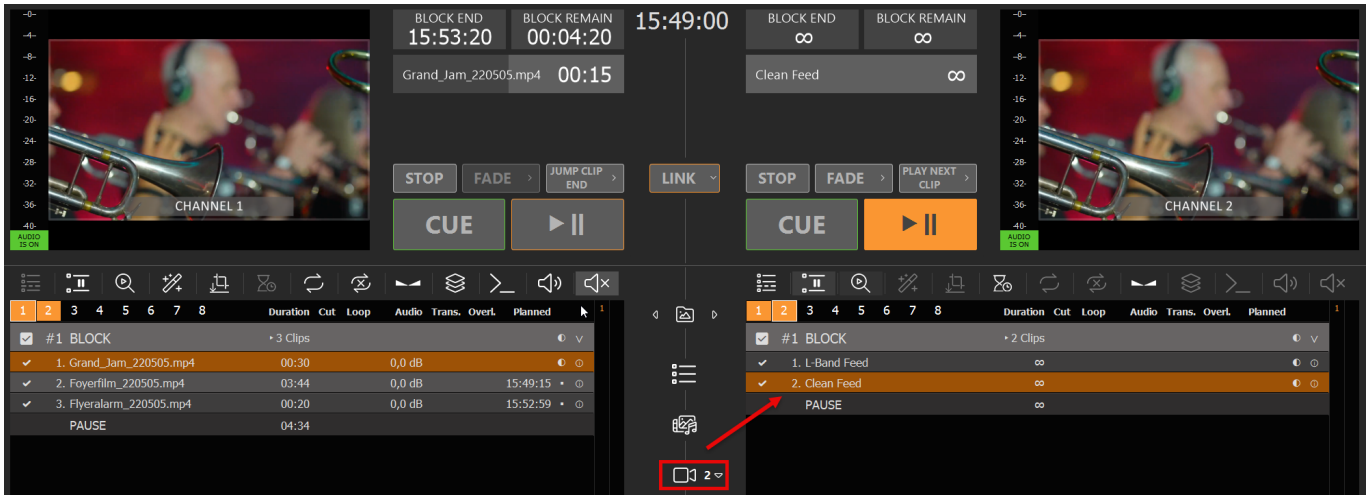


Add the Clean Feed for Switching

Since our Final Output will run over Channel 2, we also want our Clean Feed to be selectable in the Channel 2 Playlist. For this, we can simply copy the Channel without the Output Scaler. Use another Input and set Channel 1 as Source:



Then also add that Input to Channel 2. I already renamed both Inputs in the Playlist to be more distinguishable:

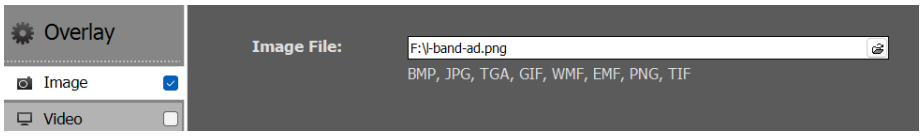


You can now quickly switch between L-Bands and Clean Feed.

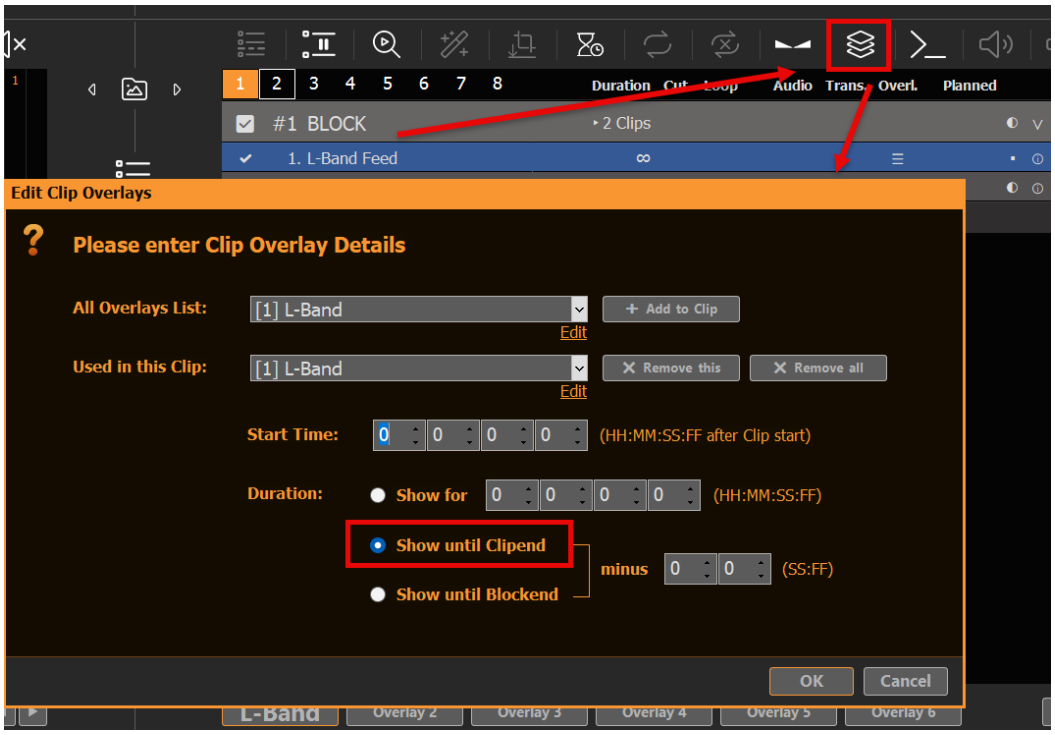
Use Overlays for L-Bands

You can use PLAYDECK Overlays for your L-Bands. For simplicity, we use this transparent PNG over the Video signal. But these can be more complex. See this article on how to create Overlay groups and fade them together.

Click on any empty Overlay Button, then add our Sample PNG:

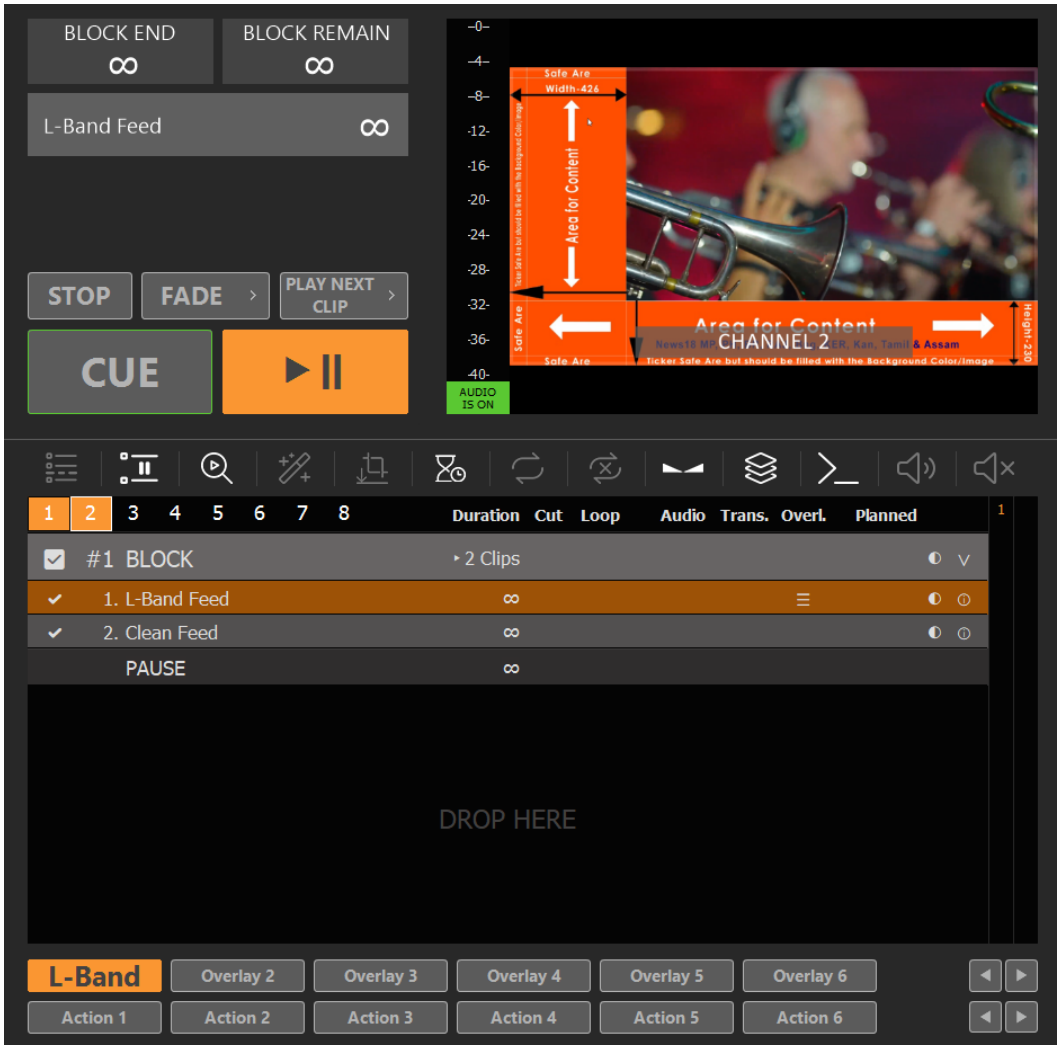


We now want our Overlay to ONLY play with the Playlist Clip for the “L-Band Feed” Input. For this we select the L-Band Clip and click the Overlay Icon, then add our new Overlay for the whole duration of the Clip:



Our L-Band Overlay will now automatically start together with the L-Band

Feed:



Note: Overlays that are assigned to Clips always have a slight reaction delay. This can be overcome by finetuning Playlist and Overlay Fade Times. You could also split the L-Band Feed and Clean Feed to Channel 2 and 3 and use an external Mixer for Transitions.

Using SCTE for Inputs and Outputs

This article will explain how you can use SCTE for your projects. PLAYDECK supports SCTE-35, which are exclusively used in Streams and SCTE-104, which are exclusively used in SDI Device signals.

Scroll down for SCTE-104. We also recommend to follow-up with this article on sending SCTE-35 to Streaming Server.

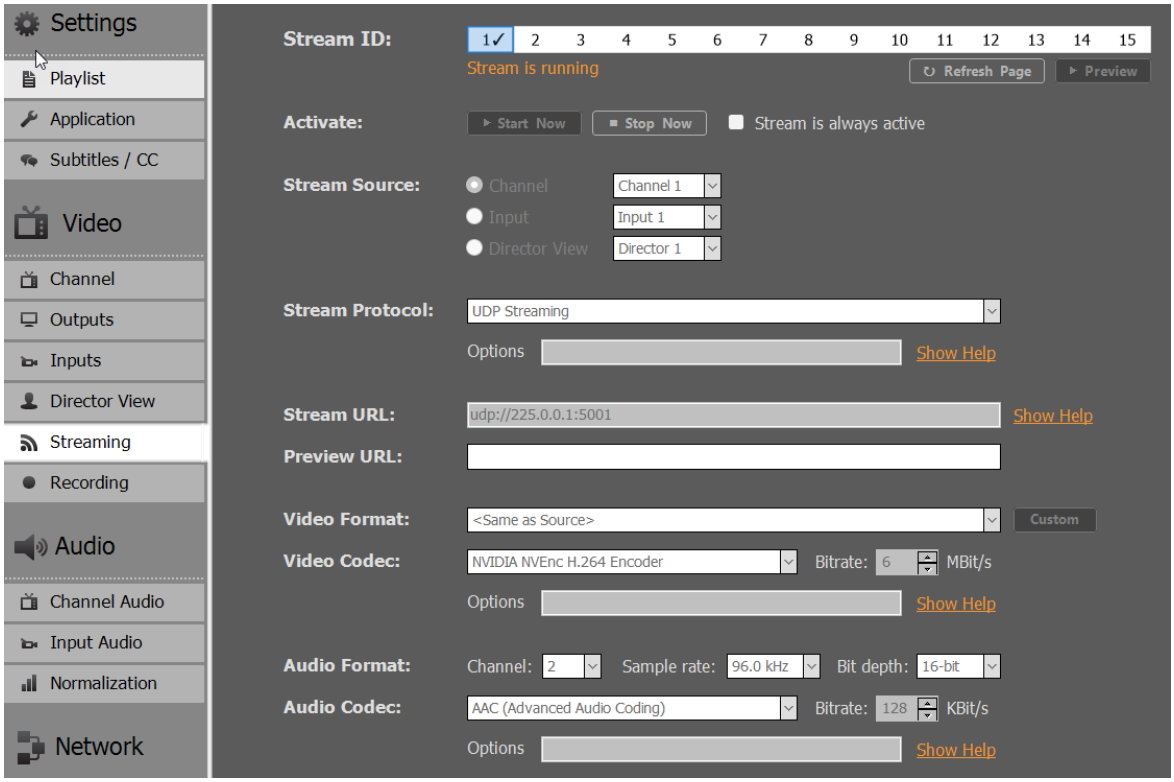
Using SCTE-35 in Streams

SCTE-35 Facts

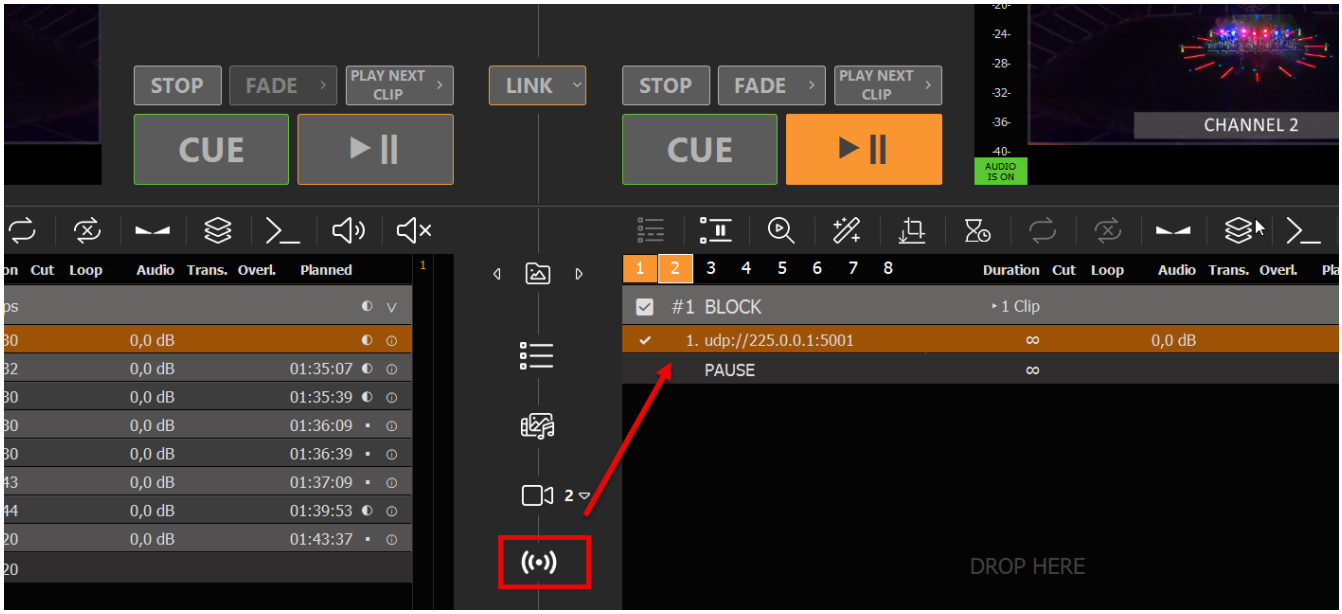
- Supported for Stream Inputs as well as Stream Outputs. Tested Stream Formats are UDP, DVB and SRT. The selection of the Video Codec does not influence SCTE transport. SCTE transport is always active and does not need to be enabled somewhere.
- PLAYDECK can detect and display incoming SCTE-35 Marker. All incoming and outgoing Marker will be written to an Event Log.
- PLAYDECK can forward incoming SCTE-35 Marker from Stream Input to Stream Output. Cross-Forwarding between UDP, DVB and SRT is supported, e.g. receiving SRT and forwarding to UDP.

SCTE-35 Testdrive

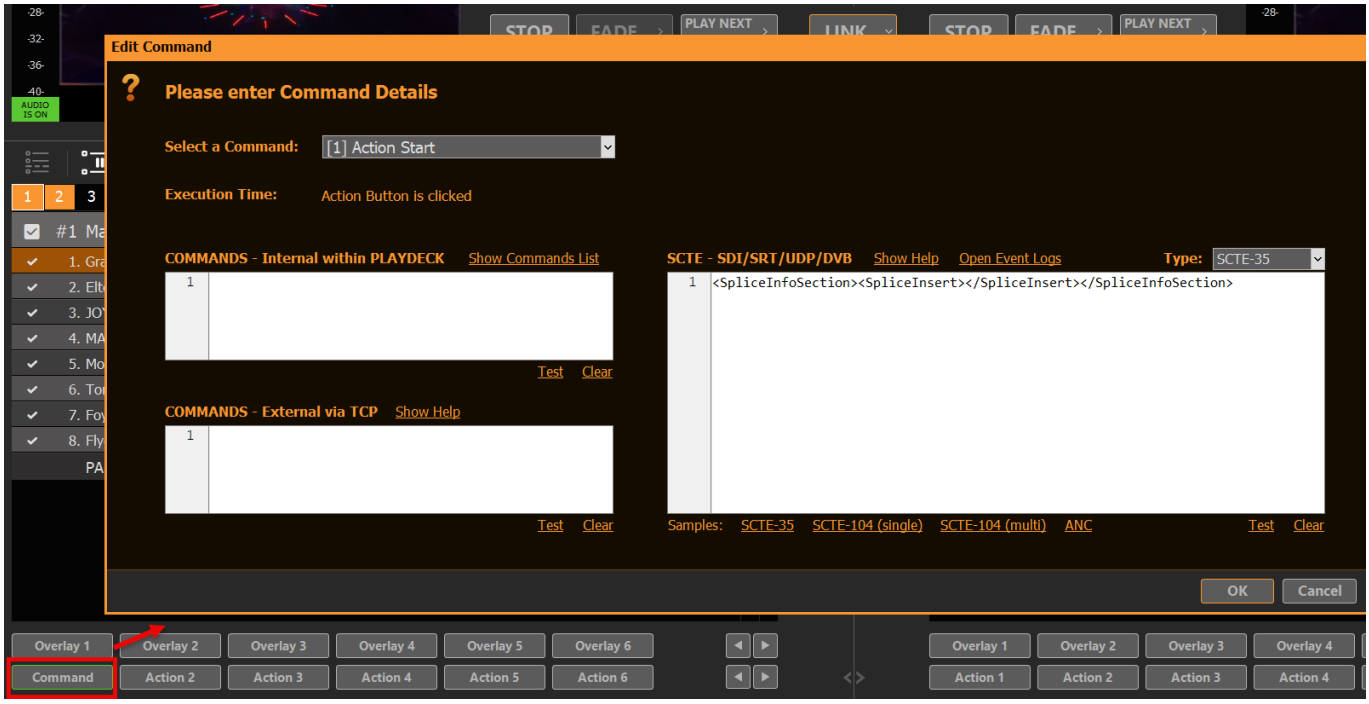
Let us dive into those topics a bit. As a testrun we setup our own UDP signal loop. We stream Channel 1 to “udp://225.0.0.1:5001” via UDP:



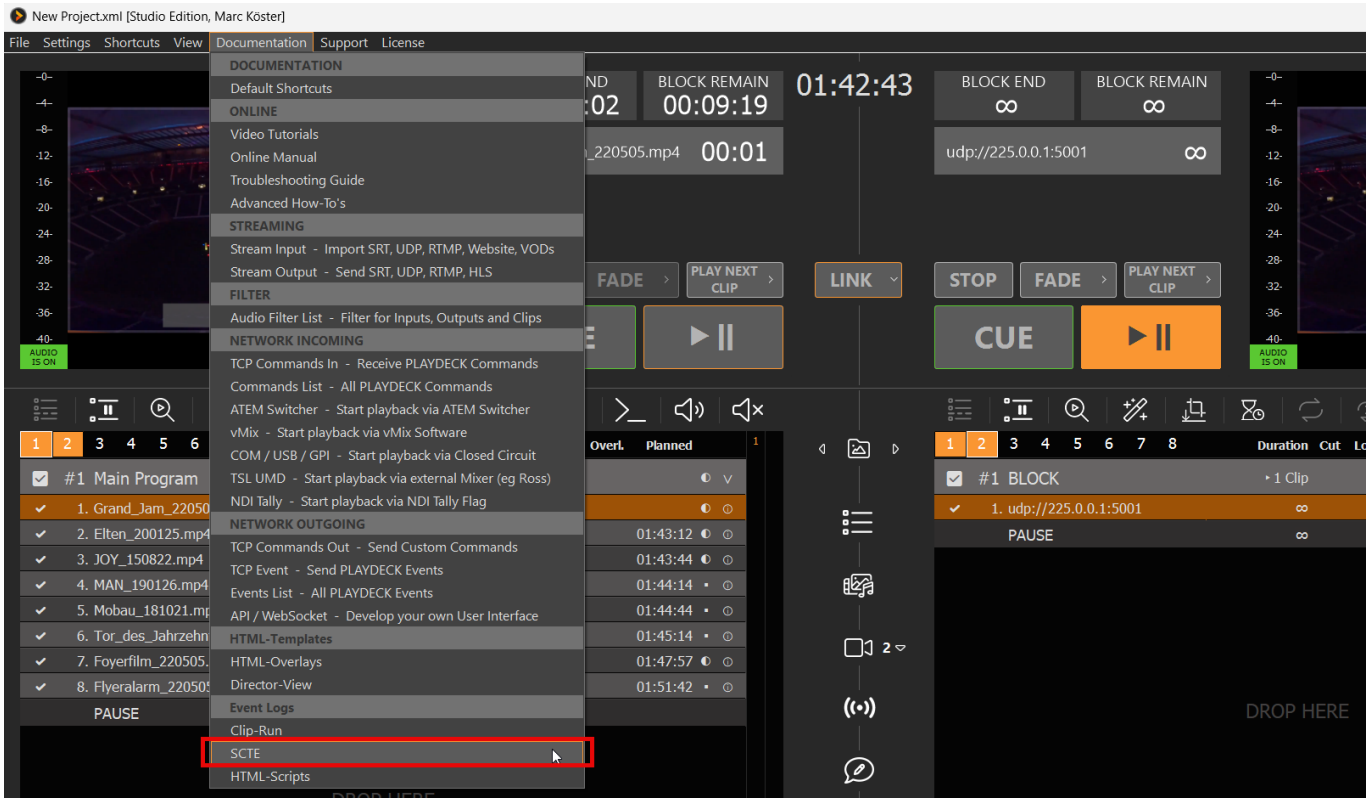
And play the Stream in Channel 2:



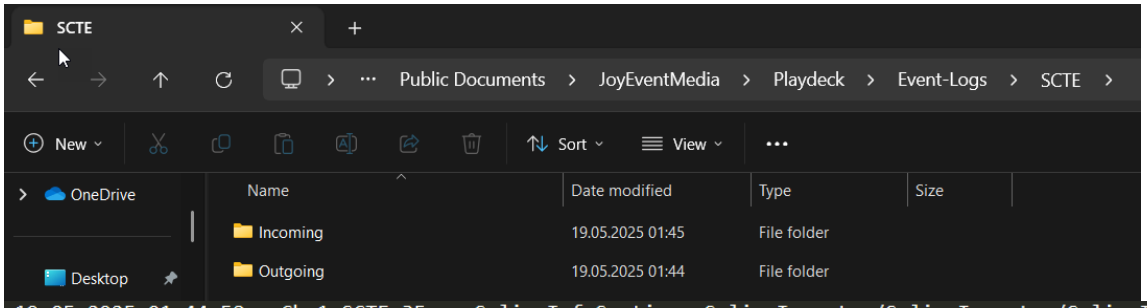
Now we only need a way to send a SCTE-35 marker in Channel 1. So we setup a new Command Action and use the smallest detectable SCTE-35 Marker (of course you can use any other):
<SpliceInfoSection><SpliceInsert></SpliceInsert></SpliceInfoSection>



Make sure Channel 1 and Channel 2 are playing, then hit the Action Button. Then open the SCTE Event Log over the Main Menu > Documentation > SCTE:

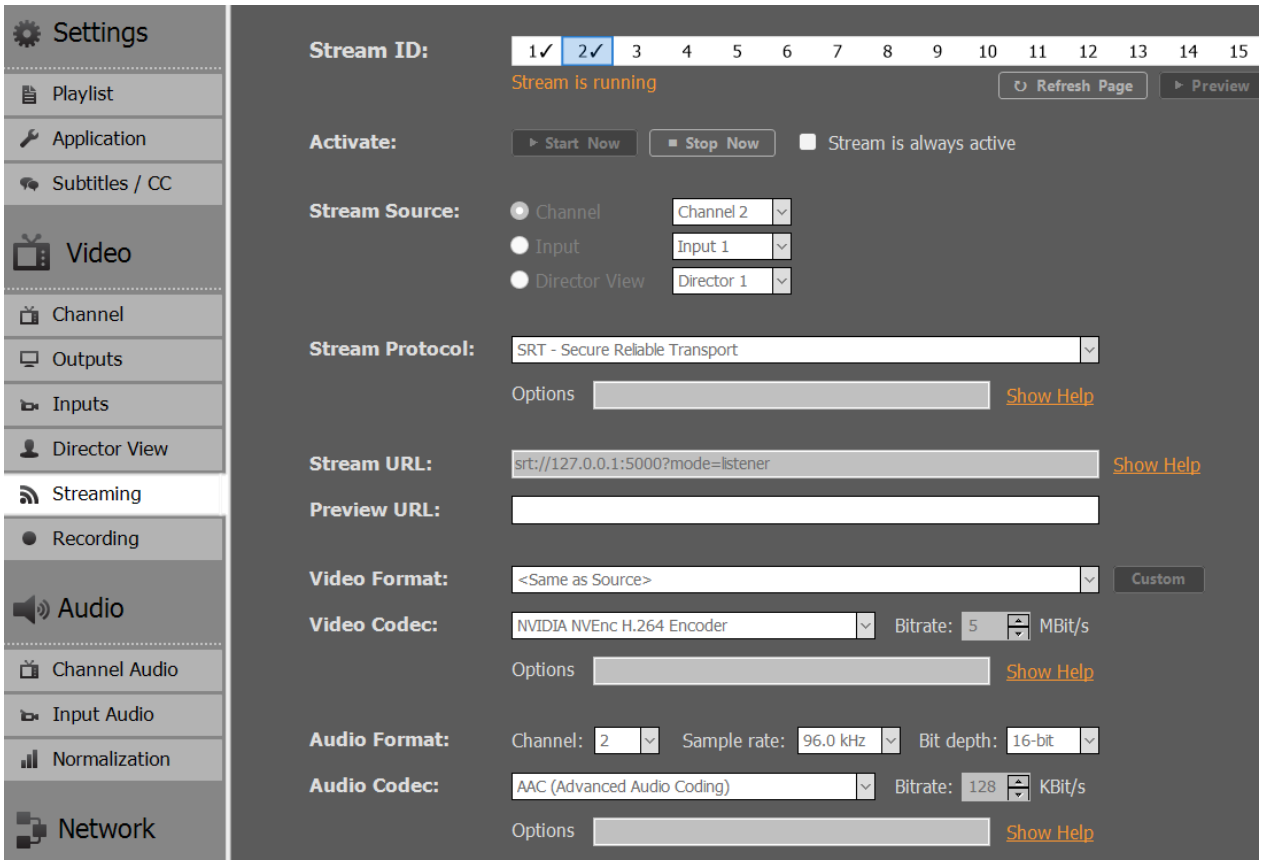


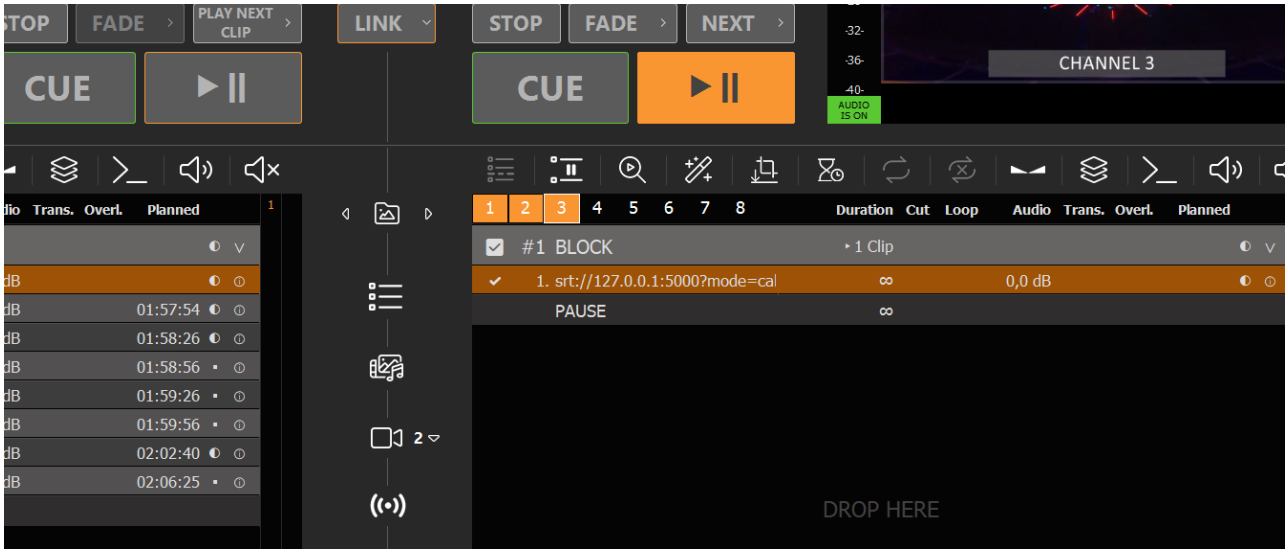
You will see the outgoing Marker of your Action Button Click on Channel 1, and the detected incoming Marker on Channel 2:




```
19.05.2025 01:44:58 - Ch 1 SCTE-35 - <SpliceInfoSection><SpliceInsert></SpliceInsert></SpliceInfoSection>

19.05.2025 01:45:02 - Ch 2 SCTE-35 - <SpliceInfoSection protocolVersion='0' ptsAdjustment='0' tier='0'>     <
SpliceInsert spliceEventId='0' spliceEventCancelIndicator='0' outOfNetworkIndicator='0' program_splice_flag='0'
spliceImmediateFlag='0' uniqueProgramId='0' availNum='0' availsExpected='0'/>  </SpliceInfoSection>
```

We even go one step further and setup ANOTHER Stream from Channel 2 to Channel 3 to test the SCTE Forwarding and Cross-over Detection (UDP to SRT). We send Channel 2 to “srt://127.0.0.1:5000?mode=listener” and receive in Channel 3 at “srt://127.0.0.1:5000?mode=caller”:





If we now press our Action Button to send the SCTE-35 Marker, we detect it also in Channel 3:

```
19.05.2025 01:45:02 - Ch 2 SCTE-35 - <SpliceInfoSection protocolVersion='0' ptsAdjustment='0' tier='0'>     <
SpliceInsert spliceEventId='0' spliceEventCancelIndicator='0' outOfNetworkIndicator='0' program_splice_flag='0'
spliceImmediateFlag='0' uniqueProgramId='0' availNum='0' availsExpected='0'/>  </SpliceInfoSection>
19.05.2025 01:54:46 - Ch 2 SCTE-35 - <SpliceInfoSection protocolVersion='0' ptsAdjustment='0' tier='0'>     <
SpliceInsert spliceEventId='0' spliceEventCancelIndicator='0' outOfNetworkIndicator='0' program_splice_flag='0'
spliceImmediateFlag='0' uniqueProgramId='0' availNum='0' availsExpected='0'/>  </SpliceInfoSection>
19.05.2025 01:54:48 - Ch 3 SCTE-35 - <SpliceInfoSection protocolVersion='0' ptsAdjustment='0' tier='0'>     <
SpliceInsert spliceEventId='0' spliceEventCancelIndicator='0' outOfNetworkIndicator='0' program_splice_flag='0'
spliceImmediateFlag='0' uniqueProgramId='0' availNum='0' availsExpected='0'/>  </SpliceInfoSection>
```

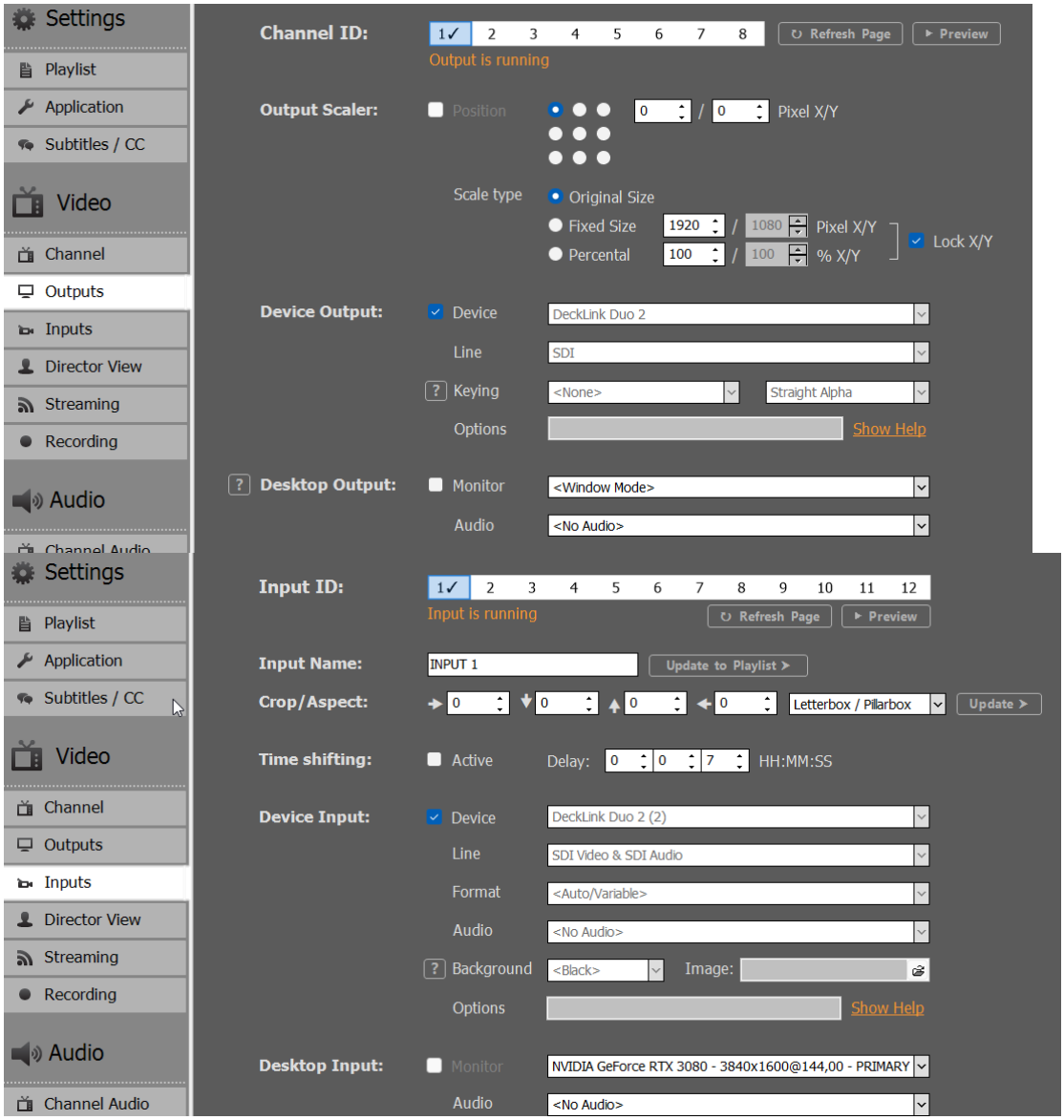
Using SCTE-104 in SDI

SCTE-104 Facts

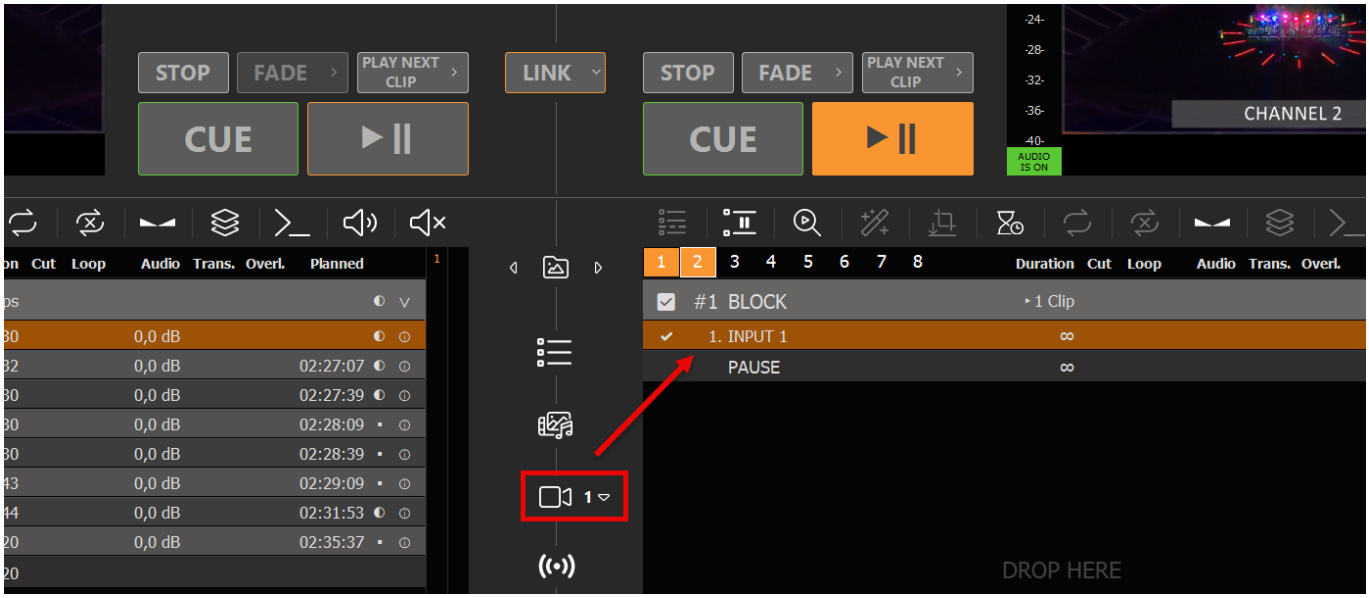
- Supported for SDI Device Inputs as well as SDI Device Outputs, e.g. Blackmagic DeckLink. SCTE transport is always active and does not need to be enabled somewhere.
- PLAYDECK can detect and display incoming SCTE-104 Marker. All incoming and outgoing Marker will be written to an Event Log.
- PLAYDECK can forward incoming SCTE-104 Marker from SDI Input to SDI Output.

SCTE-104 Testdrive

Let us dive into those topics a bit. As a testrun we setup our own SDI signal loop. We use our DeckLink Duo 2 and an SDI Loop Cable:

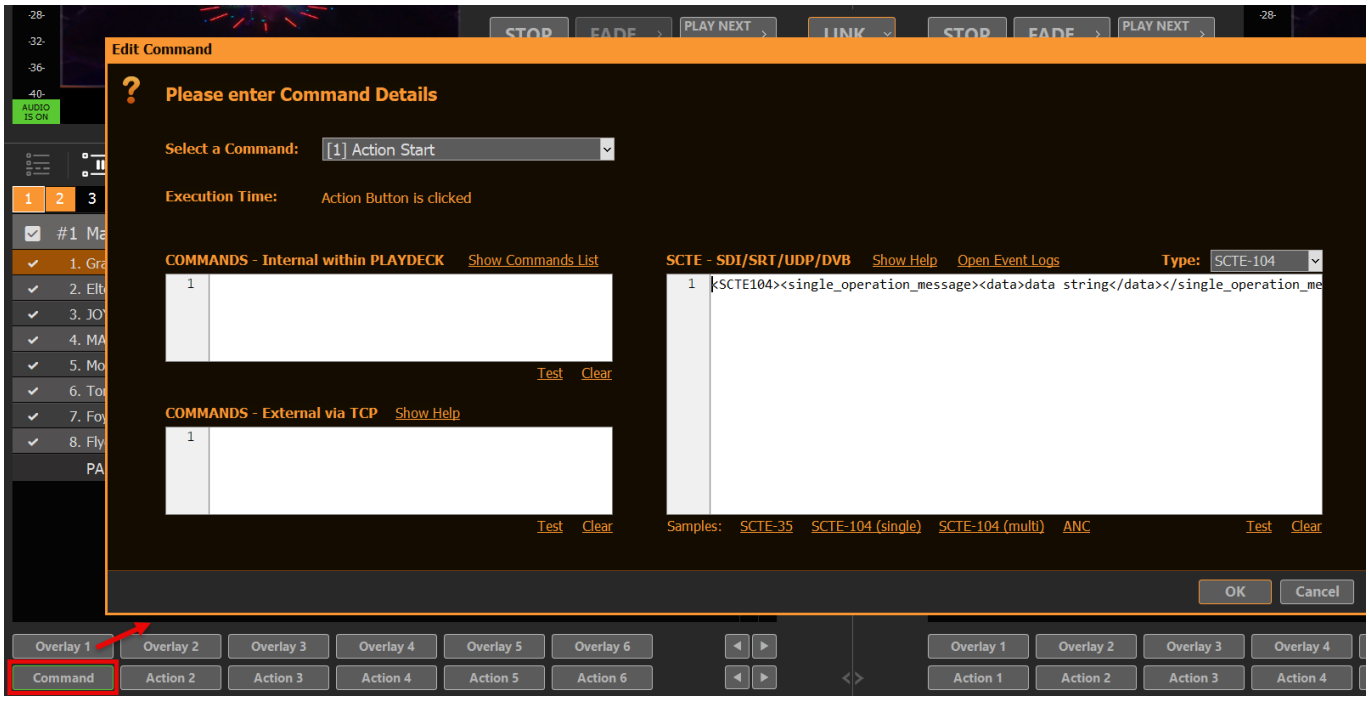


And play the SDI Input in Channel 2:

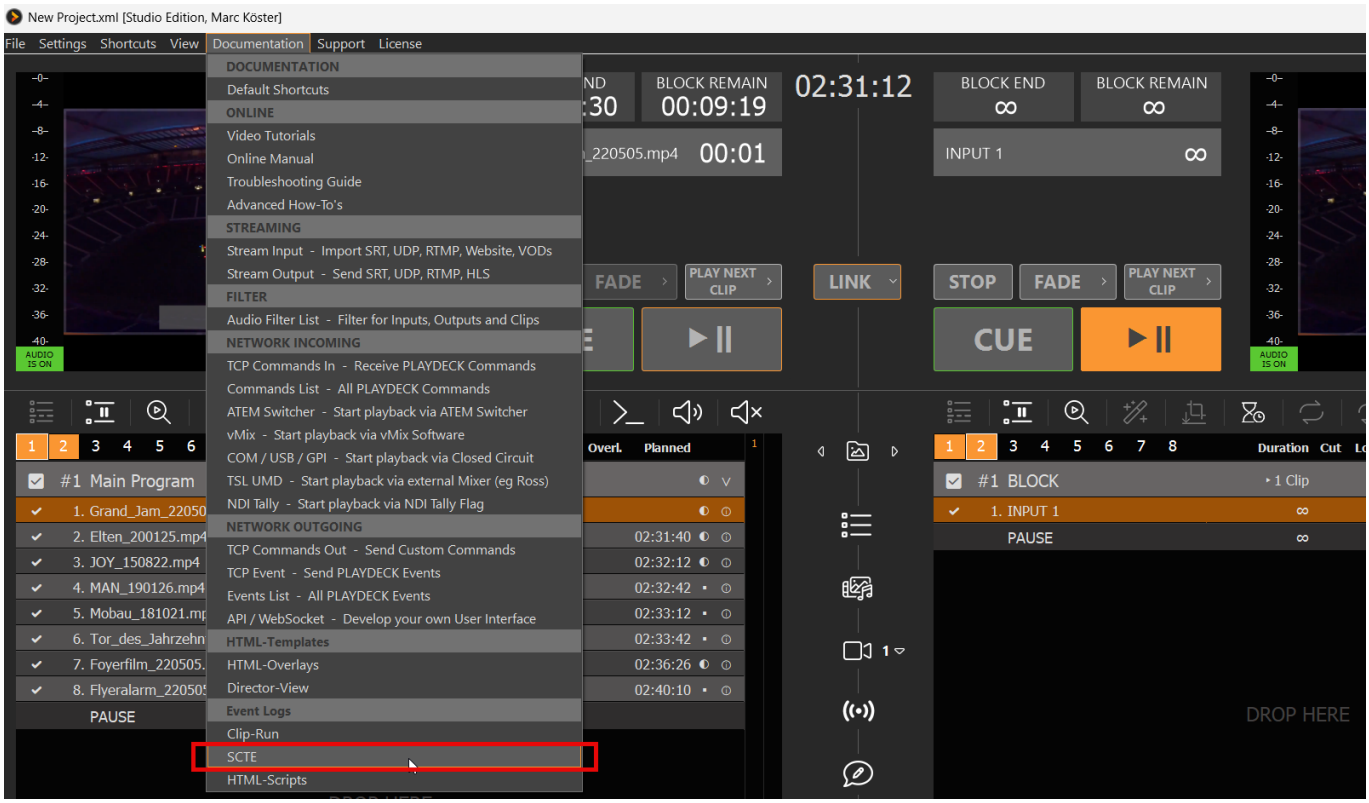


Now we only need a way to send a SCTE-104 marker in Channel 1. So we setup a new Command Action and use the smallest detectable SCTE-104 Marker (of course

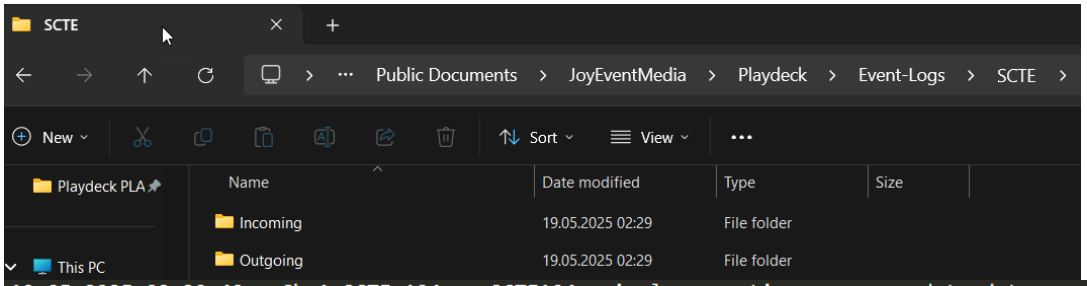
you can use any other):
<SCTE104><single_operation_message><data>data string</data></single_operation_message></SCTE104>



Make sure Channel 1 and Channel 2 are playing, then hit the Action Button. Then open the SCTE Event Log over the Main Menu > Documentation > SCTE:

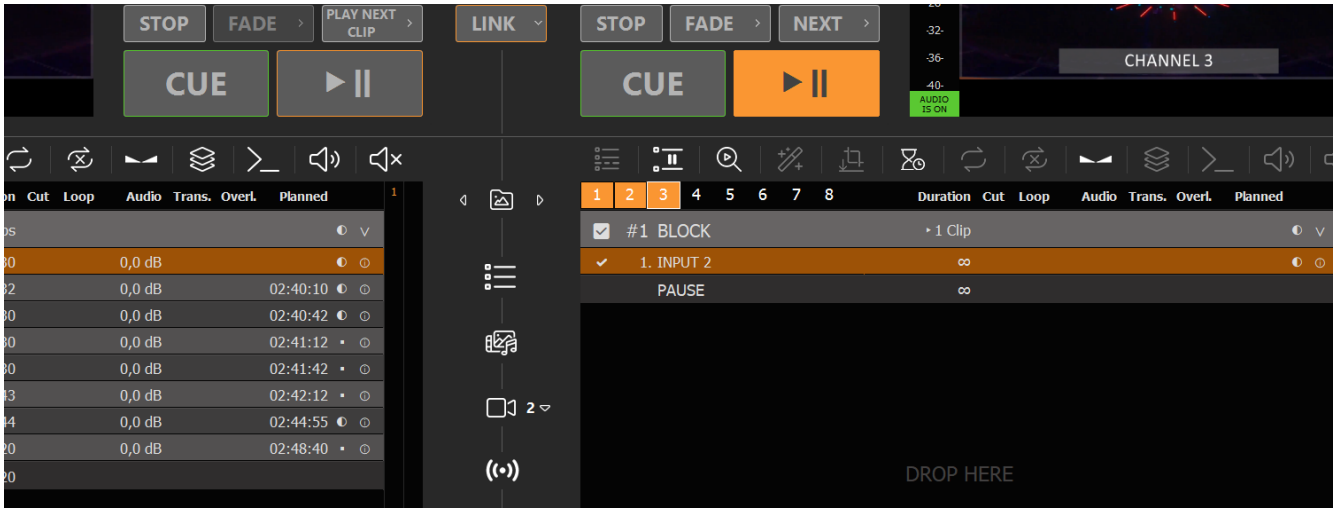


You will see the outgoing Marker of your Action Button Click on Channel 1, and the detected incoming Marker on Channel 2:




```
19.05.2025 02:29:49 - Ch 1 SCTE-104 - <SCTE104><single_operation_message><data>data
string</data></single_operation_message></SCTE104>

19.05.2025 02:29:49 - Ch 2 SCTE-104 - <single_operation_message>    <opID>0</opID>      <messageSize>
20</messageSize>   <result>0</result>      <result_extension>0</result_extension>      <protocol_version>
0</protocol_version>      <AS_index>0</AS_index>      <message_number>0</message_number>      <DPI_PID_index>
0</DPI_PID_index>    <data>          <general_response_data/>    </data>  </single_operation_message>
```

We even go one step further and setup ANOTHER SDI Loop from Channel 2 to Channel 3 to test the SCTE Forwarding:



If we now press our Action Button to send the SCTE-104 Marker, we detect it also in Channel 3:

```
19.05.2025 02:29:49 - Ch 2 SCTE-104 - <single_operation_message>    <opID>0</opID>      <messageSize>
20</messageSize>   <result>0</result>      <result_extension>0</result_extension>      <protocol_version>
0</protocol_version>      <AS_index>0</AS_index>      <message_number>0</message_number>      <DPI_PID_index>
0</DPI_PID_index>    <data>          <general_response_data/>    </data>  </single_operation_message>
19.05.2025 02:40:28 - Ch 2 SCTE-104 - <single_operation_message>    <opID>0</opID>      <messageSize>
20</messageSize>   <result>0</result>      <result_extension>0</result_extension>      <protocol_version>
0</protocol_version>      <AS_index>0</AS_index>      <message_number>0</message_number>      <DPI_PID_index>
0</DPI_PID_index>    <data>          <general_response_data/>    </data>  </single_operation_message>
19.05.2025 02:40:28 - Ch 3 SCTE-104 - <single_operation_message>    <opID>0</opID>      <messageSize>
20</messageSize>   <result>0</result>      <result_extension>0</result_extension>      <protocol_version>
0</protocol_version>      <AS_index>0</AS_index>      <message_number>0</message_number>      <DPI_PID_index>
0</DPI_PID_index>    <data>          <general_response_data/>    </data>  </single_operation_message>
```

Sending SCTE-35 to Stream Server

SCTE Marker are mainly used to insert Ads into the Video Output by signalling the Receiver (e.g. Streaming Server) at what time to insert Ads and with what Duration. This allows 3rd Party Systems to dynamically pick Ads from a Database. You can read more about SCTE-35 in general on this website.

We also recommend our general article on SCTE.

You can attach SCTE Marker to 4 different Objects in PLAYDECK:
– Clips (any Position)
– Blocks (Start and End)
– Overlays (Show and Hide)
– Actions Buttons

Instead of providing your with predefined SCTE Marker, we picked a different approach: You have to write your own SCTE Marker. This could mean a bit of a learning curve, but in the end, it is much more flexible. The advantage is, that you will always be compatible to all receiver. For gathering Clip duration and other “variables” we provide certain “placeholder”.

SCTE-35 Marker Examples

Let’s look at some SCTE-35 Sample. It represents the most used form of SCTE-35 called “Splicing”.

Note: The SCTE-35 Format presented here works best with Nimble Streamer (see below). Depending on your Stream Server, the Format might be different. Ask your Provider.

```
<SpliceInfoSection>
  <SpliceInsert
    spliceEventId="4157"
    outOfNetworkIndicator="1"
    spliceImmediateFlag="1">
    <Program />
    <BreakDuration
      autoReturn="1"
      duration="{clipduration}" />
  </SpliceInsert>
</SpliceInfoSection>
```

This will insert an Ad and return to your Program automatically after the given duration. The duration will be set automatically by PLAYDECK for your current running Clip. This can be exchanged with {blockduration} or a manual value. If you pick a manual value for duration, the formula is: CLIP DURATION multiplied by 90000 and rounded. The reason behind this is, that 90000 is default Timescale most SCTE Systems use. Example: The Clip is 12.4 seconds,

so the SCTE duration would be 1116000.

Instead of sending a duration, you could also send 2 seperate SCTE Marker to Start/End the Ad. Those are called CUE-OUT (leaving your Program for Ad) and CUE-IN (return to your Program).

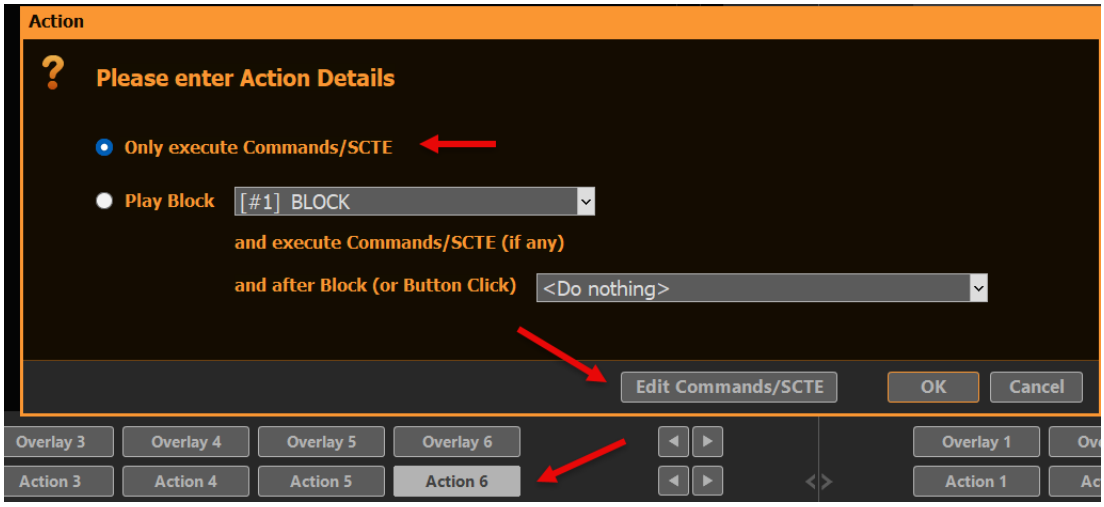
```
<SpliceInfoSection>
  <SpliceInsert
    spliceEventId="4157"
    outOfNetworkIndicator="1"
    spliceImmediateFlag="1">
    <Program />
  </SpliceInsert>
</SpliceInfoSection>

<SpliceInfoSection>
  <SpliceInsert
    spliceEventId="4157"
    outOfNetworkIndicator="0"
    spliceImmediateFlag="1">
    <Program />
  </SpliceInsert>
</SpliceInfoSection>
```

As you most likely have observed already, the only difference is “outOfNetworkIndicator” being 1 (CUE-OUT) and 0 (CUE-IN). The SCTE System itself will pair both Marker based on the “spliceEventId”.

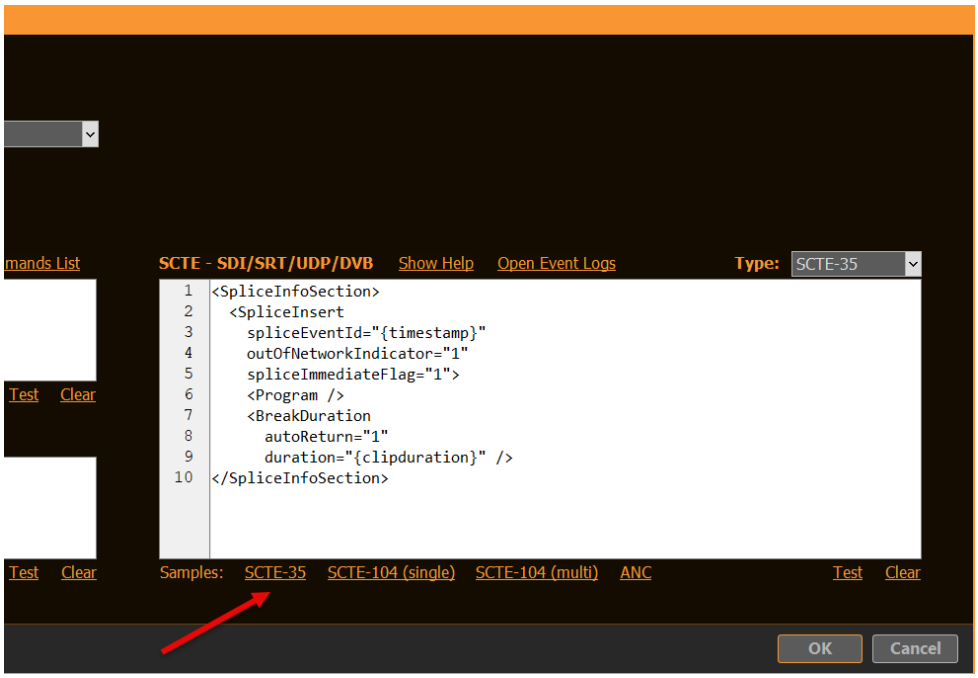
Note: The supported placeholder are: {timestamp} {timestampunix} {airtimenext} {airtimenextunix} {plannednext} {plannednextunix} {clipid} {blockid} {blockduration} {clipduration}

Test SCTE-35 Marker locally

Open PLAYDECK. You dont need to load any Clips yet. Right-Click on any ACTION Button, create a new Action and select COMMANDS/SCTE:

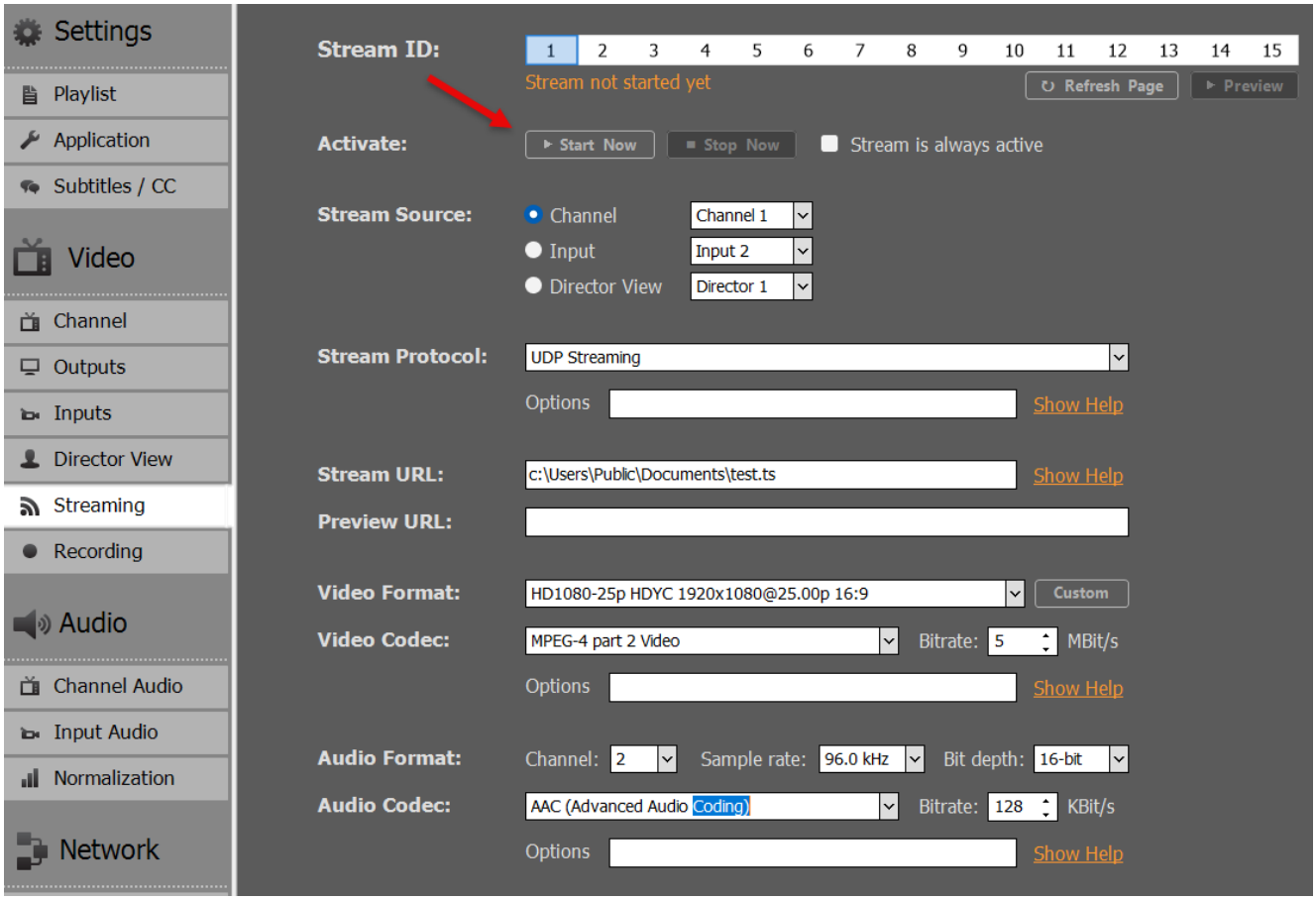


Insert the SCTE-35 Sample Tag by clicking SCTE-35. Close the Popups with OK (twice):



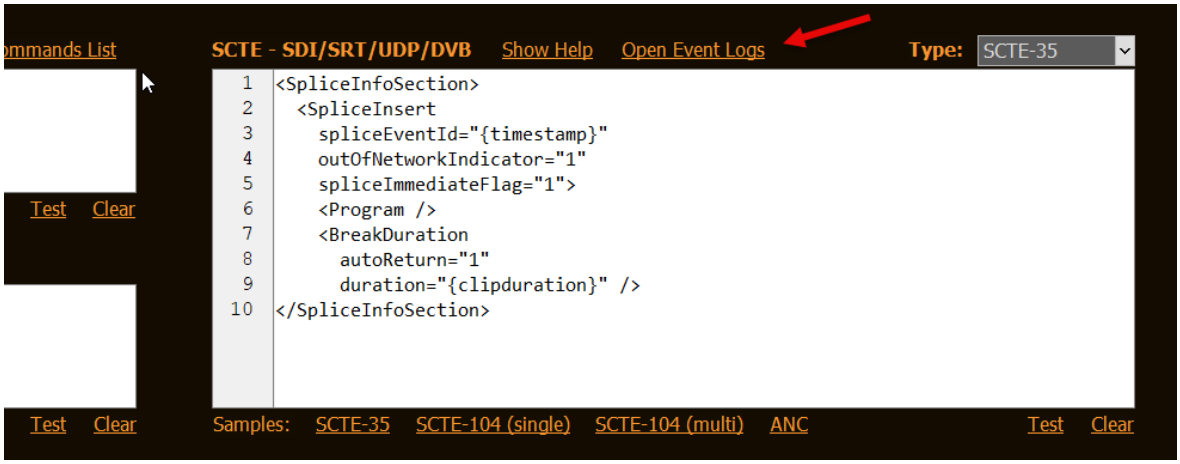
We are now able to send a SCTE-35 Tag to any compatible Stream (UDP, DVB, SRT). If you click the Action now, nothing will happen, as we dont have a Stream yet.

Let’s create a Test UDP Stream, where we simply send the Output to a File, so we don’t have to hassle with Server Stuff (yet). Copy the Settings as follows and start the Stream with START NOW.



We want to send our Sample SCTE-35 Tag to the Stream now. So close Settings and click your Action Button several Times, then go back to Settings and stop your Stream with STOP NOW.

Your SCTE-35 Tags have been logged by PLAYDECK. You can check the Logs, if you go back to your SCTE-35 Code on the Action Button and click OPEN EVENT LOGS:

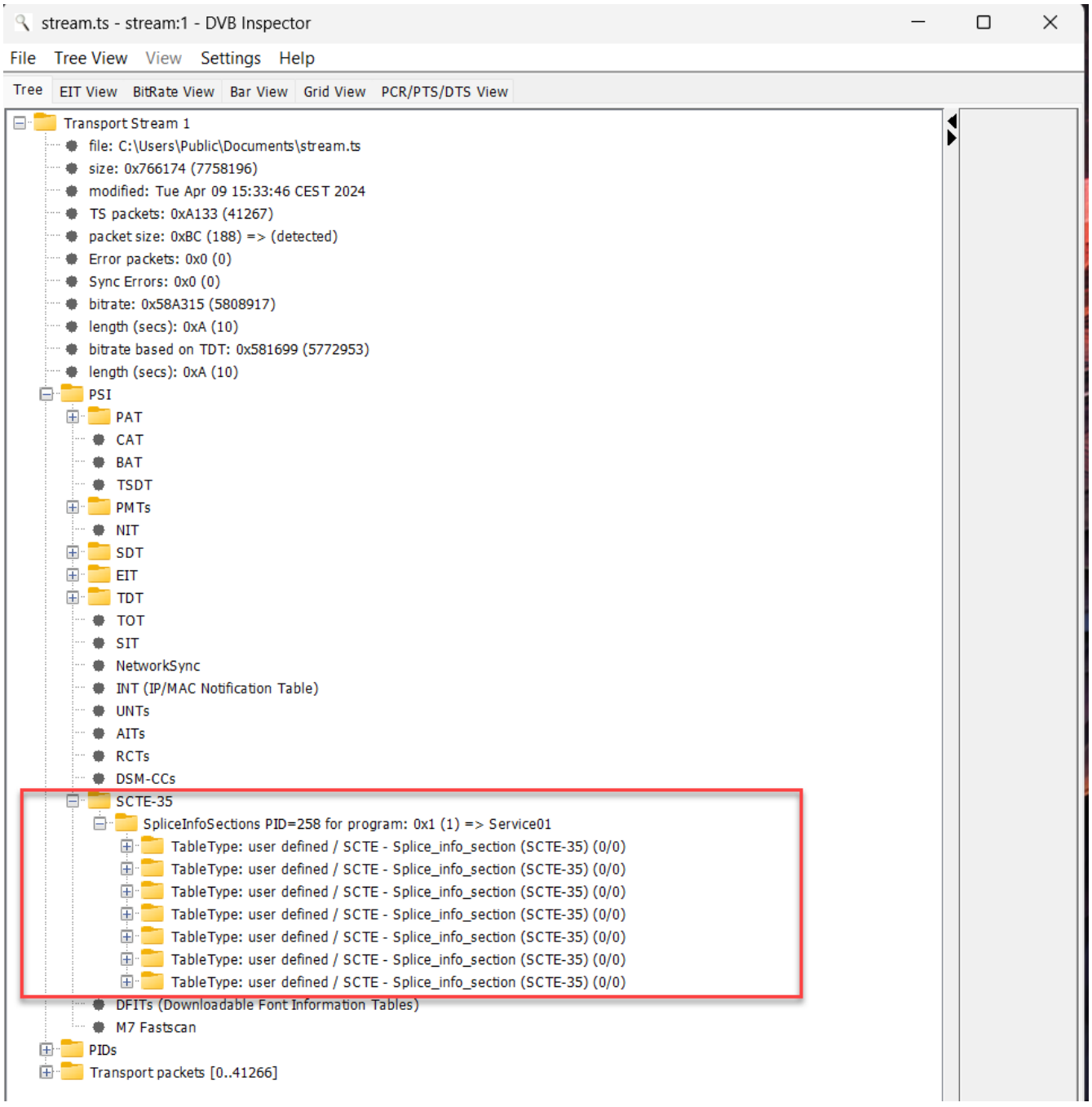


You can also make the SCTE-35 Tags visible, which are in your recorded Stream. For this, we need a separate Tool called “DVBInspector”. This Tool also needs JAVA Runtime, so download both:

- JAVA Runtime
- DVDInspector

This Tool requires JAVA Runtime. If it is not installed already, you can download here:
https://www.oracle.com/java/technologies/downloads/

Now unzip your “DVBinspector-1.18.0-dist.zip” (or similar) and start “DVBinspector-1.18.0.jar”. Then Drag+Drop your TS File onto the Application. Open the Tree as follows to view your SCTE Marker.



Test SCTE-35 Marker in Nimble Streamer

Nimble Streamer is our Streaming Service of choice. It is cost-efficient, easy to setup and maintain, but still has the depth of complexity for advanced use cases:

https://softvelum.com/nimble/

Install the 14-day Trial Version of Nimble to the same machine that PLAYDECK is running. This is not a necessity, but makes things simpler, as you don’t have to put up with a network setup. You will also be asked by Nimble to create an account with WMSPanel, which is a web based setup tool for Nimble, as the Nimble Streamer itself installs without User Interface.

INFO: For this SCTE-35 Sample to work, you DONT need to have “Nimble Live Transcoder” or “Nimble Advertizer” PlugIns, if you test this on a Live Installation. They are disabled in the Trial Installation by Default.

Once you got Nimble running, there is one more configuration you need to do manually. Open the Nimble Server Configuration file here:
c:\Program Files\Nimble Streamer\conf\nimble.conf

At the bottom of this file, add the following Lines and save the file.

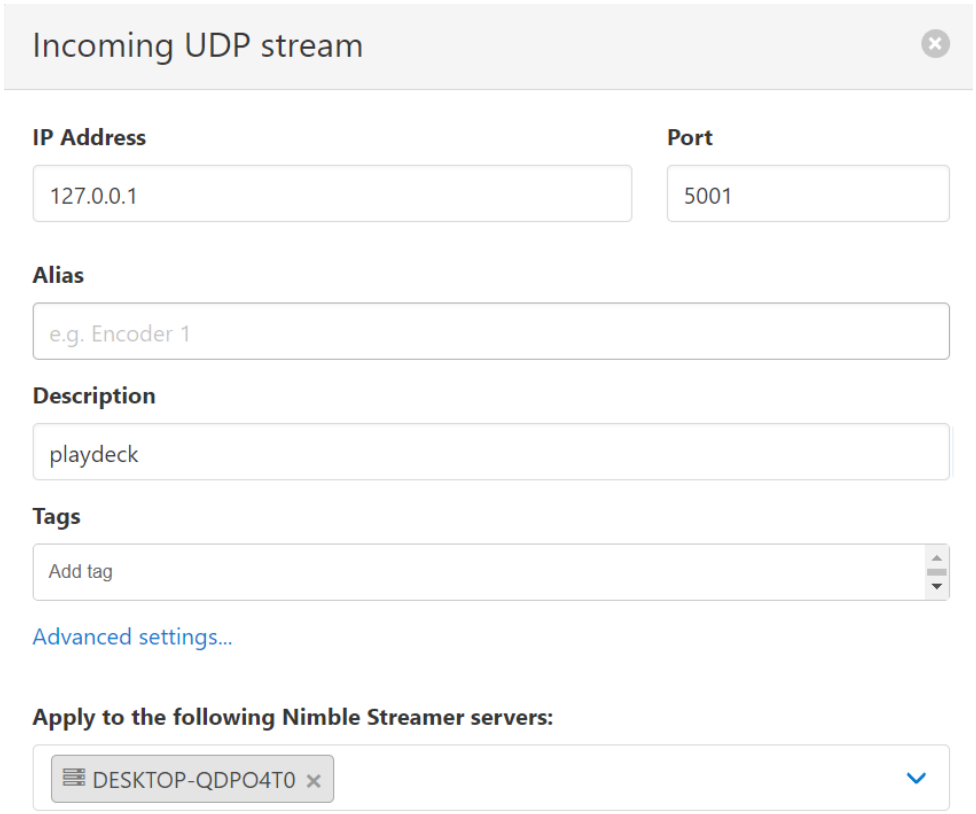
```
scte35_processing_enabled = true
scte35_forwarding_enabled = true
hls_ad_scte35_forwarding_enabled = true
hls_ad_marker_format = cue
hls_ad_splice_out_cont_marker_enabled = true
```

Important: You need to restart Nimble now by either restarting your PC or restarting the Windows Service “Nimble Streamer” manually.

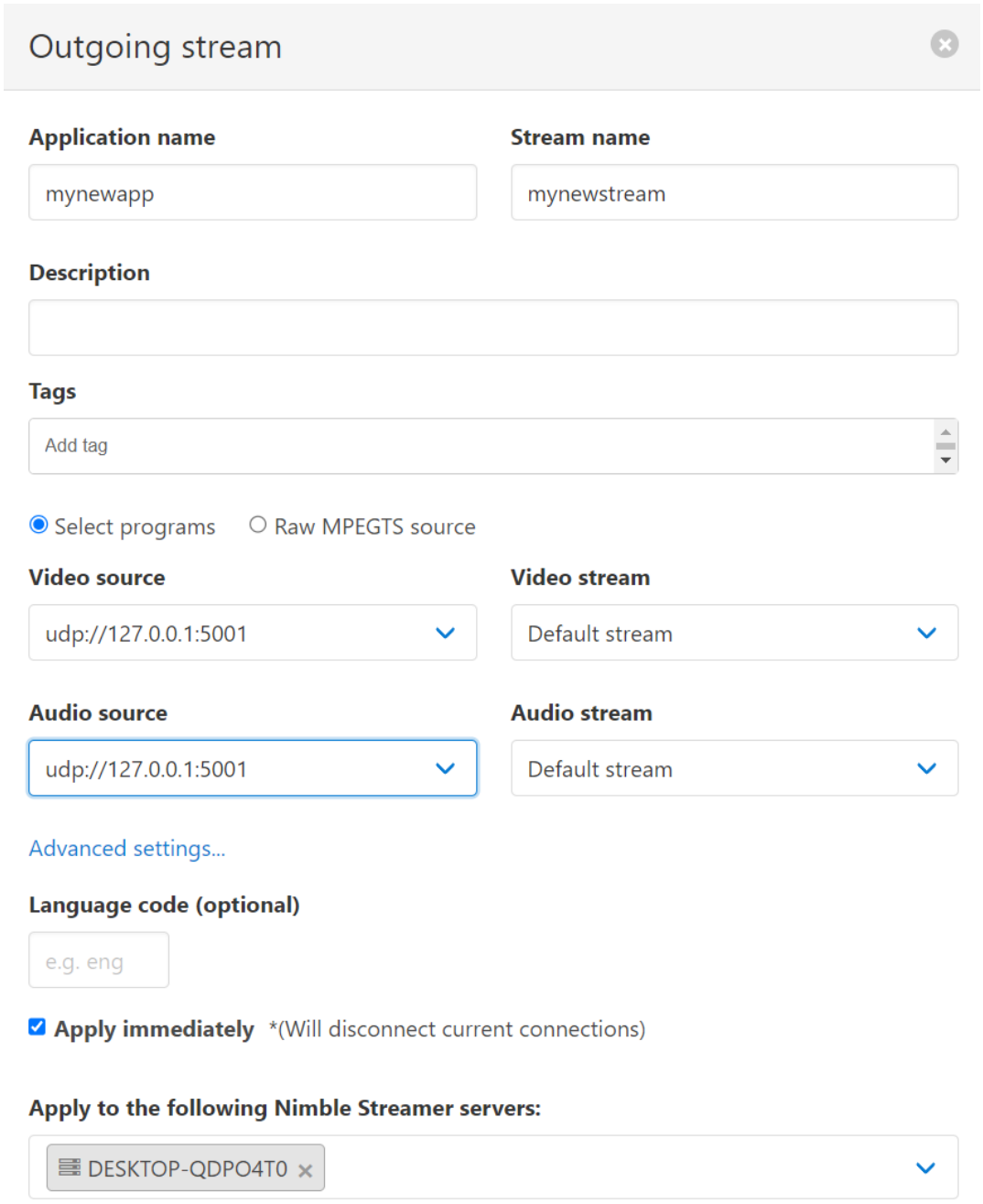
Here are the lines again, but with comments for you (dont use these):

```
scte35_processing_enabled = true  // Enabled the SCTE module in general. Needed for all other Options
scte35_forwarding_enabled = true  // Allows passing through the SCTE-35 markers via Live Transcoder
hls_ad_scte35_forwarding_enabled = true  // Enables forwarding of SCTE-35 markers into HLS streams
hls_ad_marker_format = cue  // Sets Nimble to use CUE-OUT and CUE-IN markers
hls_ad_splice_out_cont_marker_enabled = true  // Enables automatic insertion of EXT-X-CUE-OUT-CONT
```

Now let’s set up your Streams in WMSPanel:
First we are going to define our incoming Stream. Goto “Nimble Streamer” in the Menu and select “MPEGTS In” and click the “Add UDP stream” Button, then add the incoming stream like this. You can use any other Port, should 5001 be in use already.

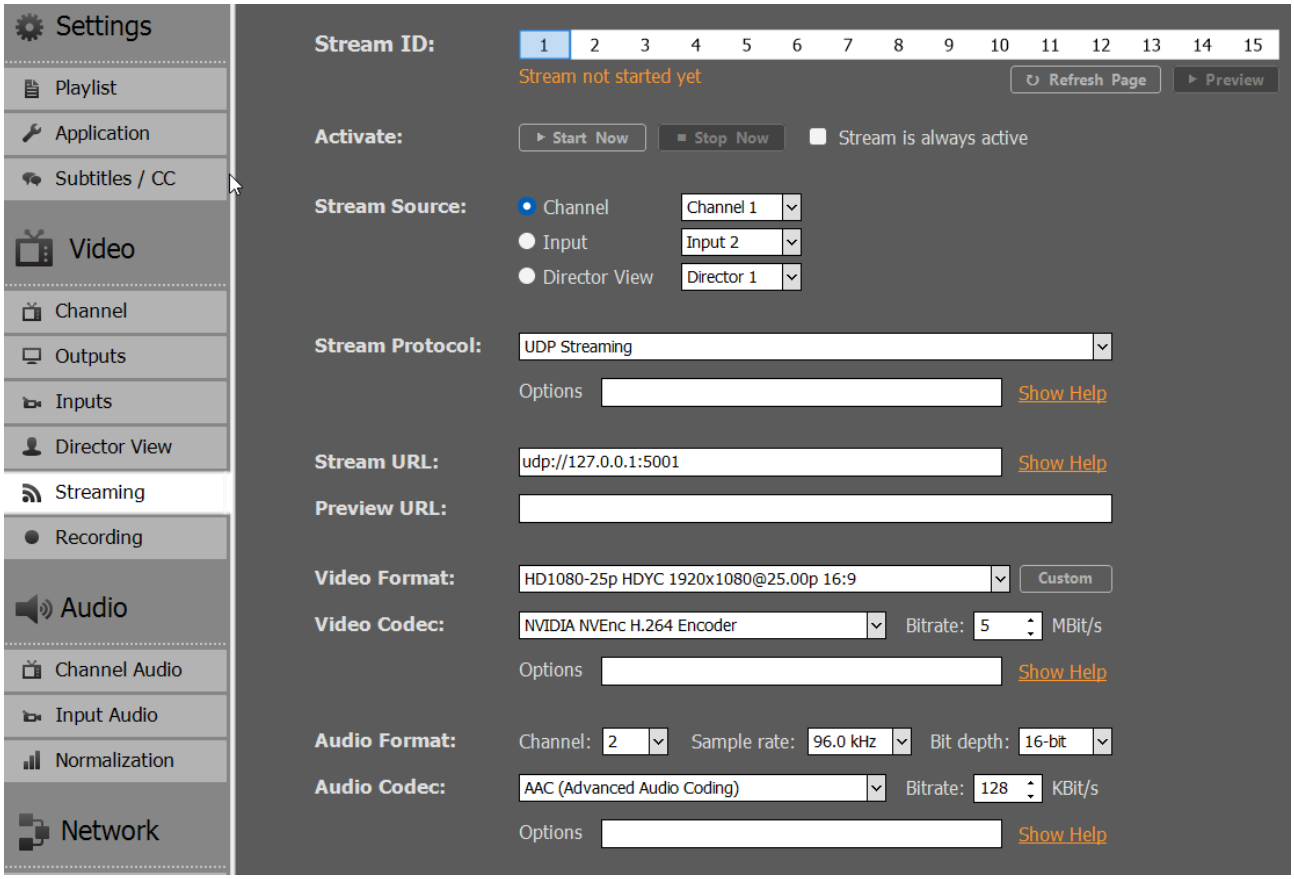


Next we are going to setup the outgoing stream format. We want to use HLS here to demonstrate the conversion of SCTE Marker from UDP to HLS. On the same page, goto “MPEGTS Out” and click the “Add outgoing stream” Button, then add like this. Make sure you select you input streeam as video and audio source.



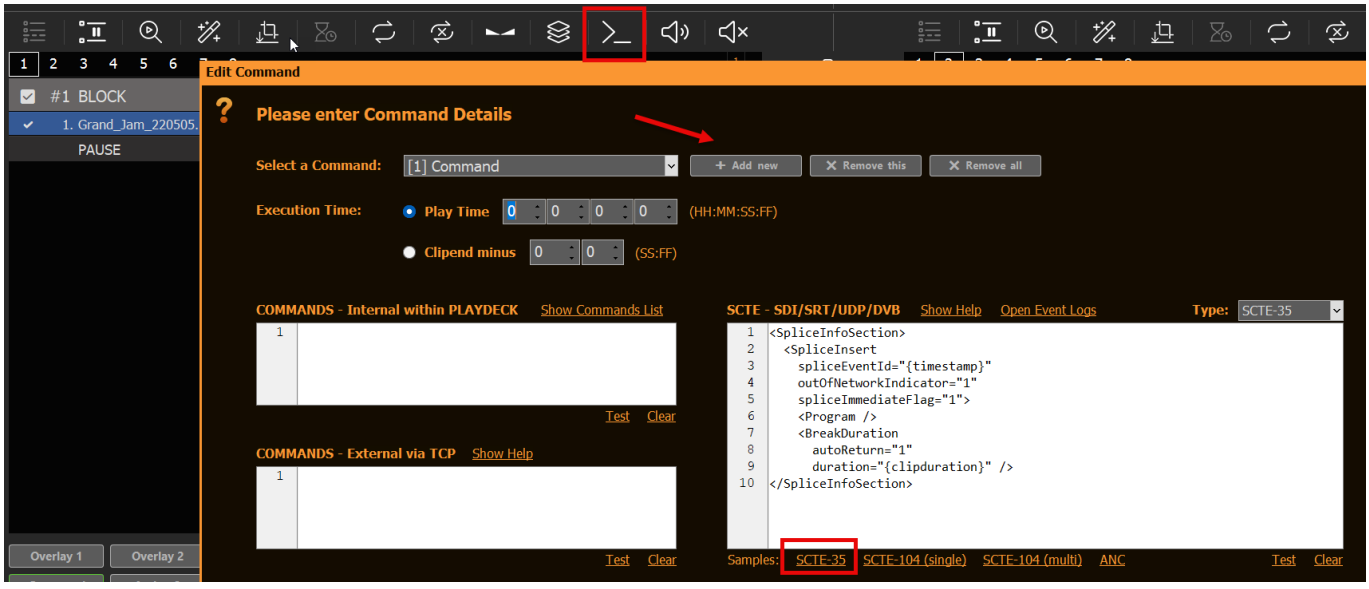
Next we continue our setup in PLAYDECK. Got to the Stream Output Settings and change the Stream URL to “udp://127.0.0.1:5001”.

INFO: You can pick any preferred Video Codec. SCTE-35 will always be send. In this case I picked H.264 to let my NVIDIA GPU do the encoding, which will lower my CPU usage.



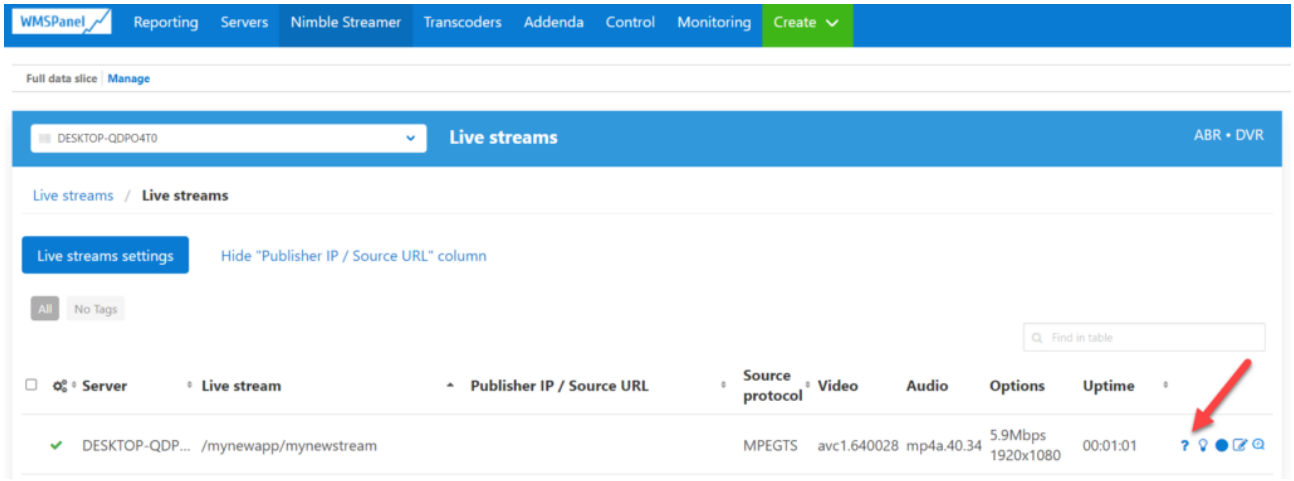
INFO: Please note, that PLAYDECK is not capable to directly stream HLS. Tho you “could” select “Apple HTTP Live Streaming” as Streaming Format and write to File e.g. “c:\Users\Public\Documents\stream.m3u8”. Then use that File to stream using any Streaming Server. But PLAYDECK is not able to write HLS SCTE Tags into the HLS Manifest file (the *.3u8 file).

Next we want PLAYDECK to automatically send SCTE, not just via a Button, but Clip related. So we insert any Clip and click the COMMANDS Icon in the Toolbar, then add a new Command, select PLAY TIME 0, then insert the SCTE-35 Sample:



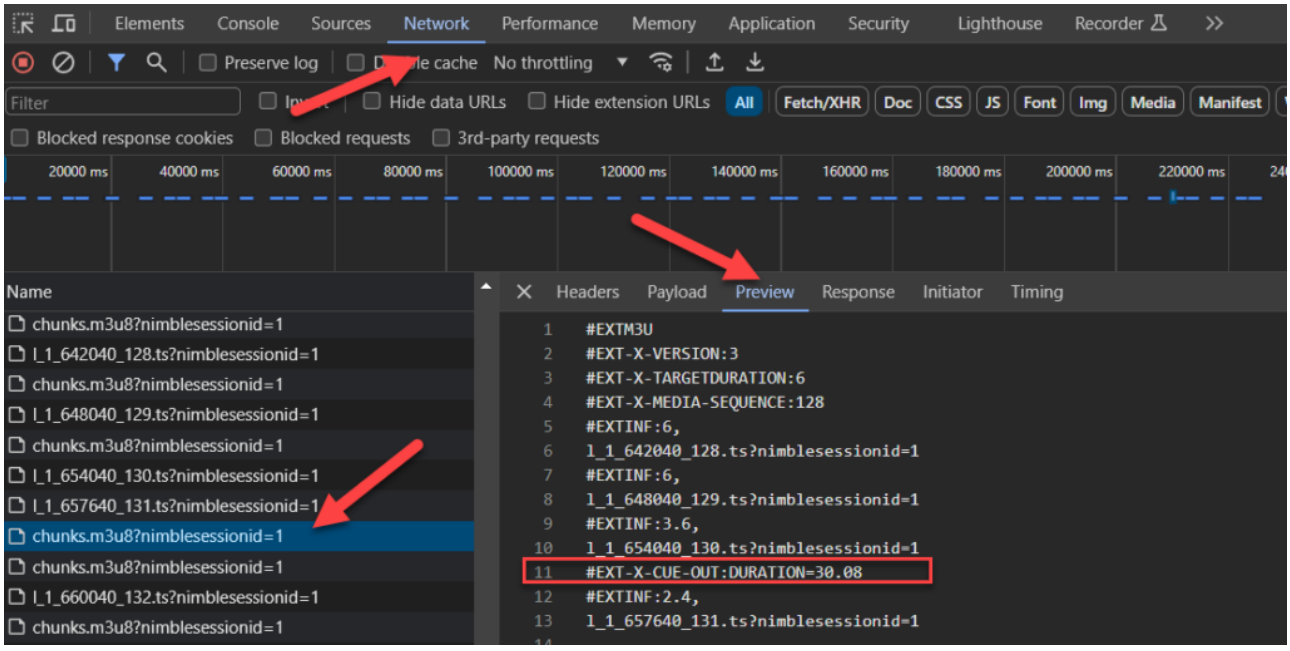
INFO: You can also add SCTE Marker to Blocks, Overlays and Action Buttons.

Now let’s get things rolling: CUE and PLAY the Clip and start the Stream in the PLAYDECK Settings. Hop over to the WMSPanel and open from the Menu “Nimble Streamer” the option “Live Streams”. You will now see 1 running Live stream. Click on it and pick the question mark on the far right like this.



From the list of “URLs to play” select the one with the “playlist.m3u8” ending and click “Copy”. Now paste this URL into any Browser, that has an HLS Player integrated or installed. For this example we use the CHROME Browser with the follwing HSL Player Extension: https://chromewebstore.google.com/detail/hls-player-m3u8-streaming/eakdijdofmnclopcffkkgmndadhbjgka

Once you posted that URL into CHROME, you will see your PLAYDECK stream playing immediatly. Now let me show you how to monitor your SCTE Marker. In Chrome press F12 to open the Developer Tools. Ontop click the “Network” Tab. You will now see several “chunks.m3u8”, which are part of you stream. Click on one of the Chunk files (also called HLS Manifest) and select “Preview” next to it. You will now see the HLS SCTE Tag. If it isnt there, click on another chunk, as we send the SCTE only on Clip Start.



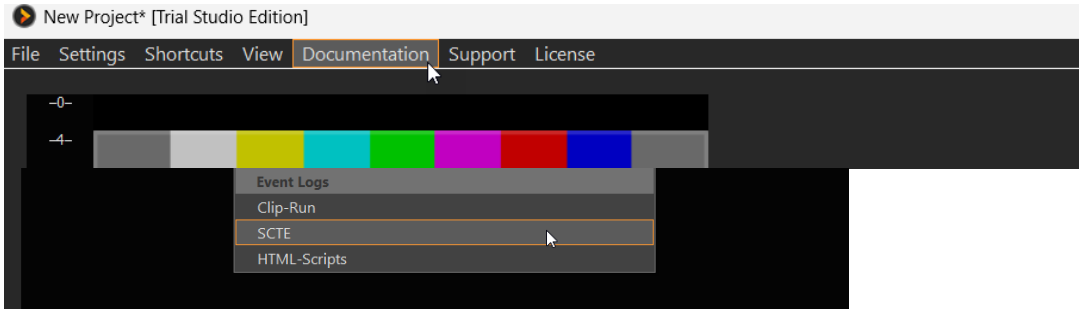
That’s it! We reached our Test Goals. Upcoming are more information which

might be relevant to dig into.

Additional Information

Logging in PLAYDECK

We log Incoming and Outgoing SCTE Tags separately. This gives you also the opportunity to test your outgoing SCTE by simply Looping your Output to an Input (SDI as well as Streams), and let PLAYDECK detect your SCTE. You can open the Log Files via the Main Menu:



Logging in Nimble Streamer

If anything goes wrong or doesnt happen as expected, a good start is to review the Nimble Log Files. The Log can be found here on Windows:
c:\Program Files\Nimble Streamer\log

Logging is enabled by default in the Nimble Trial addition, but in case you need to activate it, here are the flags for the “nimble.conf”.

```
log_access = file
logging = debug
```

Now open the log file and scan for “SCTE”. You might see something like this, which indicates you have a typo in your SCTE Marker in PLAYDECK:

```
[SCTE35] unsupported splice command type=6 for [mynewapp/mynewstream]
```

If your SCTE Marker parsed succesful in Nimble, it will look like this:

```
[SCTE35] splice_insert: event_id=4157 oon=1 immediate=1 pts_adjustment=0
pts=0 actual_pts=202593600 duration=2707200 data=0xfc....
```

Tool: TSDuck

This is a collection of tools and plugins with a large library to manipulate MPEG transport streams. With this you would be able to insert SCTE Marker outside of PLAYDECK based on an algorithm or other complex logic.

https://tsduck.io/

Tool: x9k3

This tool can also convert SCTE-35 Marker in UDP Streams or MPEG-TS Files into HLS SCTE Tags. This allows you to install a streaming server, which does not have automatic SCTE-35 to HLS conversion.

https://github.com/futzu/x9k3

Amazon EC2 Installation

PLAYDECK supports most Amazon EC2 server instances and the NVidia GPU Power they provide. This allows you to setup a cloud based infrastructure for NDI transport or other purposes.

We assume, you have a Amazon AWS Account and basic knowledge of EC2.

First, you pick a new EC2 instance, which supports a PLAYDECK installation: Windows Server with a virtual NVIDIA GPU and pre-installed NVIDIA driver: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-nvidia-driver.html#preinstalled-nvidia-driver

A typical instance would be “Windows 2019 / g4dn.xlarge” – It has a Tesla virtual NVIDIA GPU and good enough Specs to run PLAYDECK: 16 vCPUs (Intel Xeon), 64GB RAM, 1 vGPU (NVidia T4) with 16GB GPU RAM.

Now install the Instance via those Links or your EC2 Management Console.

To connect to the instance via RDP you first have to open Port 3389 in the Instance Security Settings.

Once connected, you find yourself unable to download anything via the Browser. Therefore enable Downloads like this: START Menu > Server Manager > Local Server > IE Enhanced Security Configuration > Off

Now download and install PLAYDECK as usual: https://playdeck.tv/download/

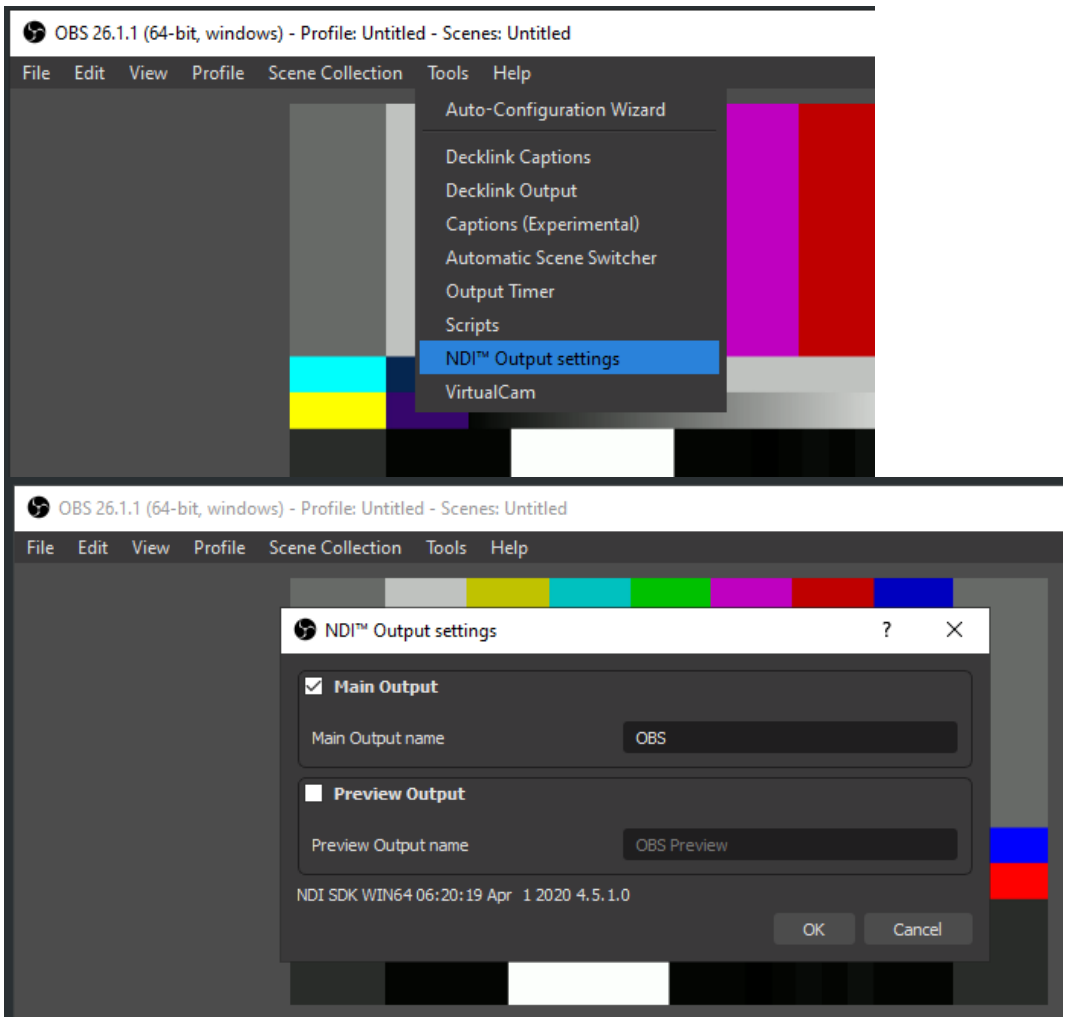
Video from/to OBS Studio

The best way to connect OBS with PLAYDECK (both ways) is NDI. Multichannel Audio is also fully supported.

Since OBS does does not deliver NDI out of the box, you need to install an additional PlugIn, which is a quick and easy process. Download the PlugIn here (scroll down and click on “distroav-6.0.0-windows-x64-Installer.exe”). It is installed like any other Windows application via Installer.

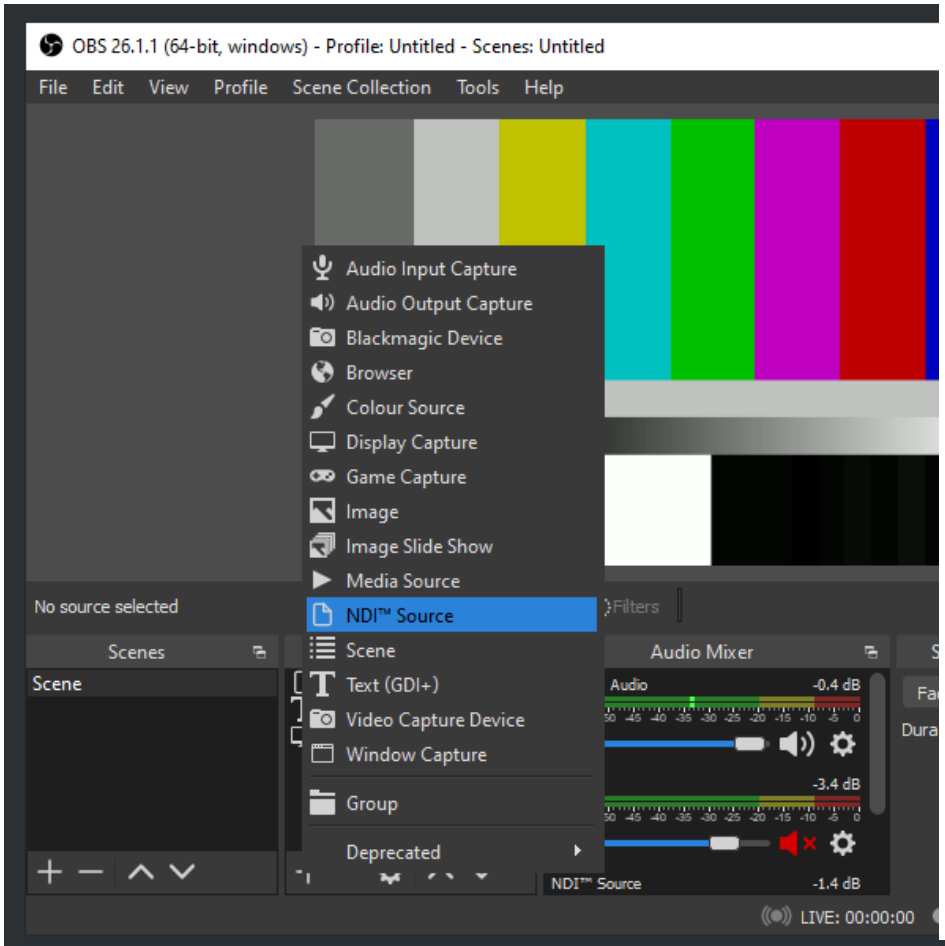
Send Video from OBS to PLAYDECK

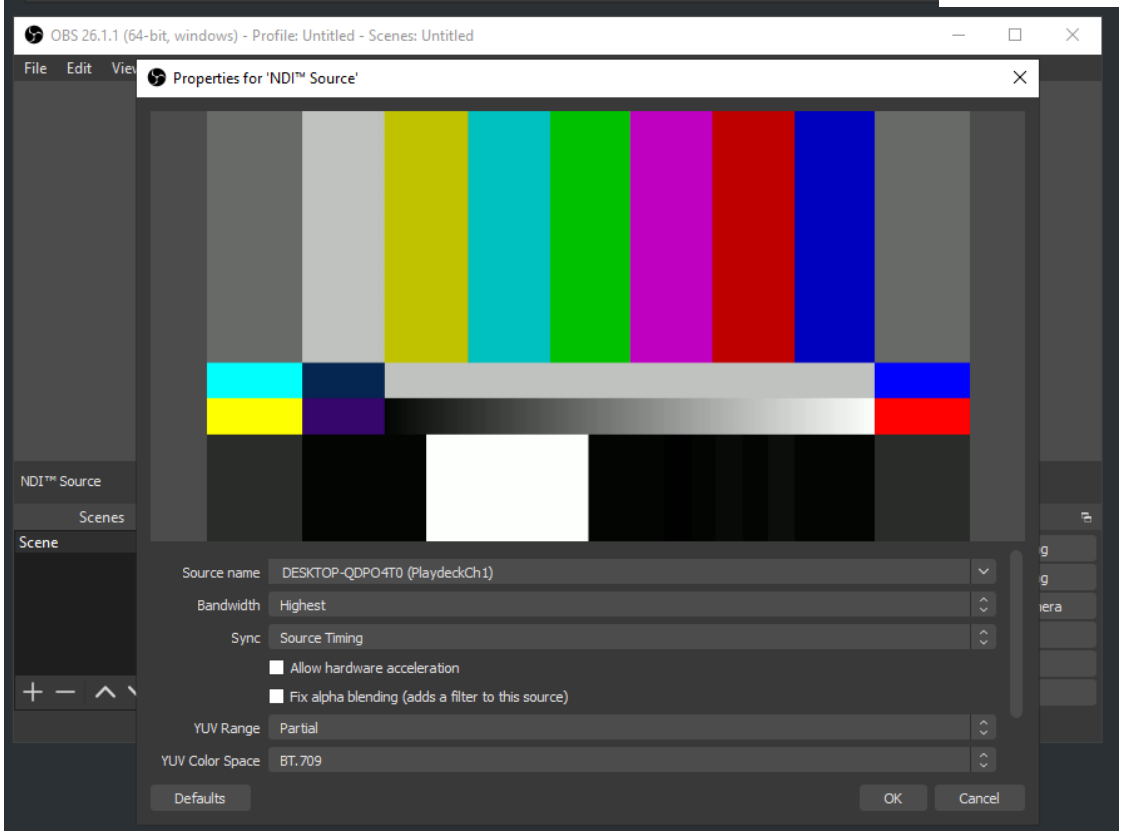
The NDI PlugIn inserts a new Option into the Tools Menu. Pick NDI Output and activate it:



Send Video from PLAYDECK to OBS

Activate NDI Output in PLAYDECK. The OBS NDI PlugIn inserts a new NDI Source, which you can add to your OBS Scene like this:


OBS 26.1.1 (64-bit, windows) - Profile: Untitled - Scenes: Untitled
File
Edit
View
Profile
Scene Collection
Tools
Help
Audio Input Capture
Audio Output Capture
Blackmagic Device
Browser
Colour Source
Display Capture
Game Capture
Image
Image Slide Show
Media Source
NDI™ Source
Scene
Text (GDI+)
Video Capture Device
Window Capture
Group
Deprecated
No source selected
Filters
Scenes
Scene
Audio Mixer
Audio
-0.4 dB
-3.4 dB
NDI™ Source
-1.4 dB
LIVE: 00:00:00

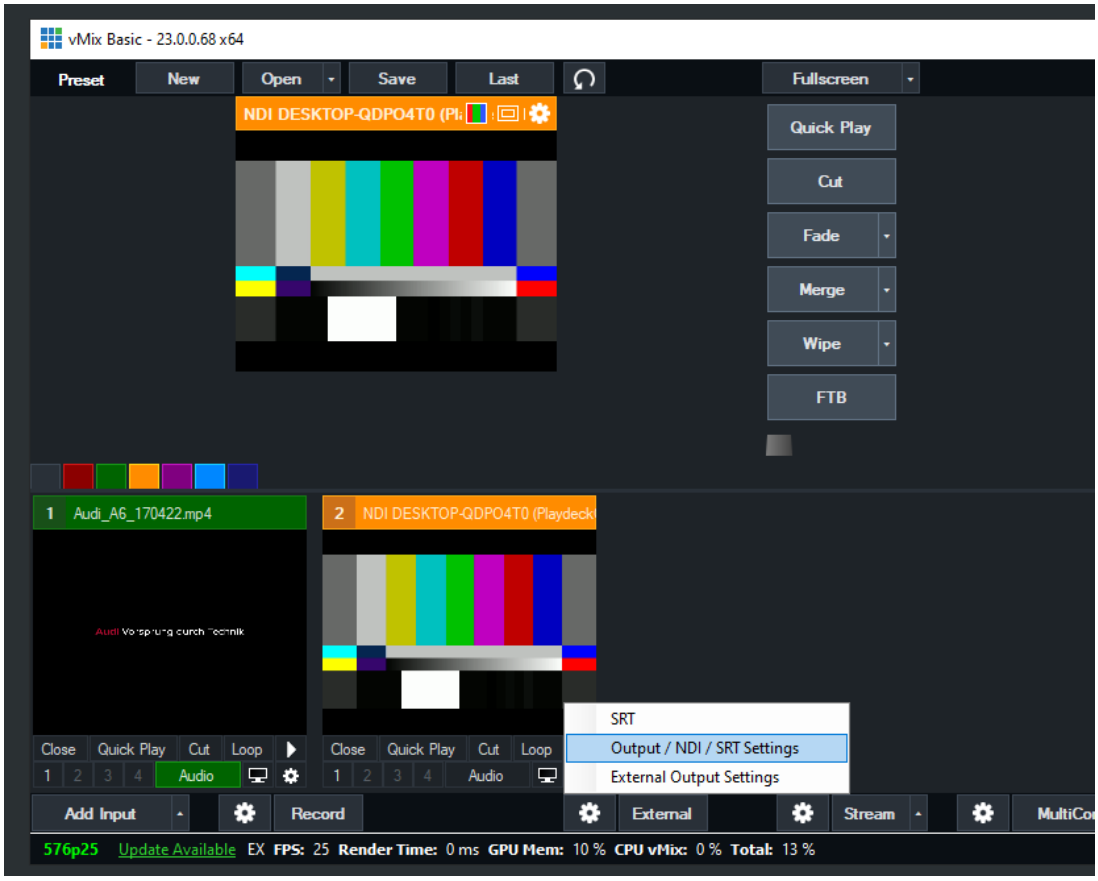

OBS 26.1.1 (64-bit, windows) - Profile: Untitled - Scenes: Untitled
File
Edit
Properties for 'NDI™ Source'
NDI™ Source
Scenes
Scene
Source name
DESKTOP-QDPO4T0 (PlaydeckCh1)
Bandwidth
Highest
Sync
Source Timing
Allow hardware acceleration
Fix alpha blending (adds a filter to this source)
YUV Range
Partial
YUV Color Space
BT.709
Defaults
OK
Cancel

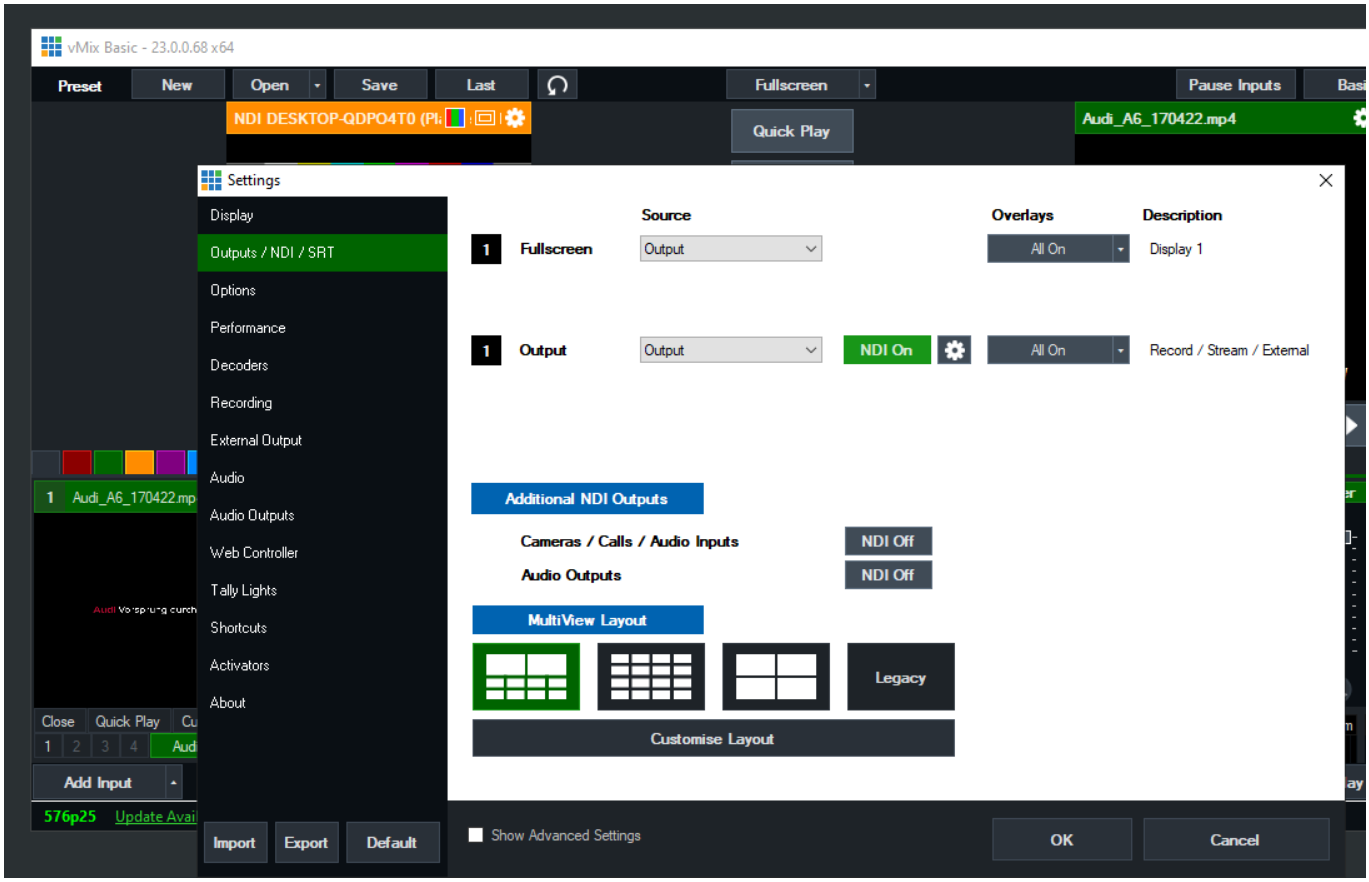
Video from/to vMix

The best way to connect vMix with PLAYDECK (both ways) is NDI. Multichannel Audio is also fully supported.

Send Video from vMix to PLAYDECK

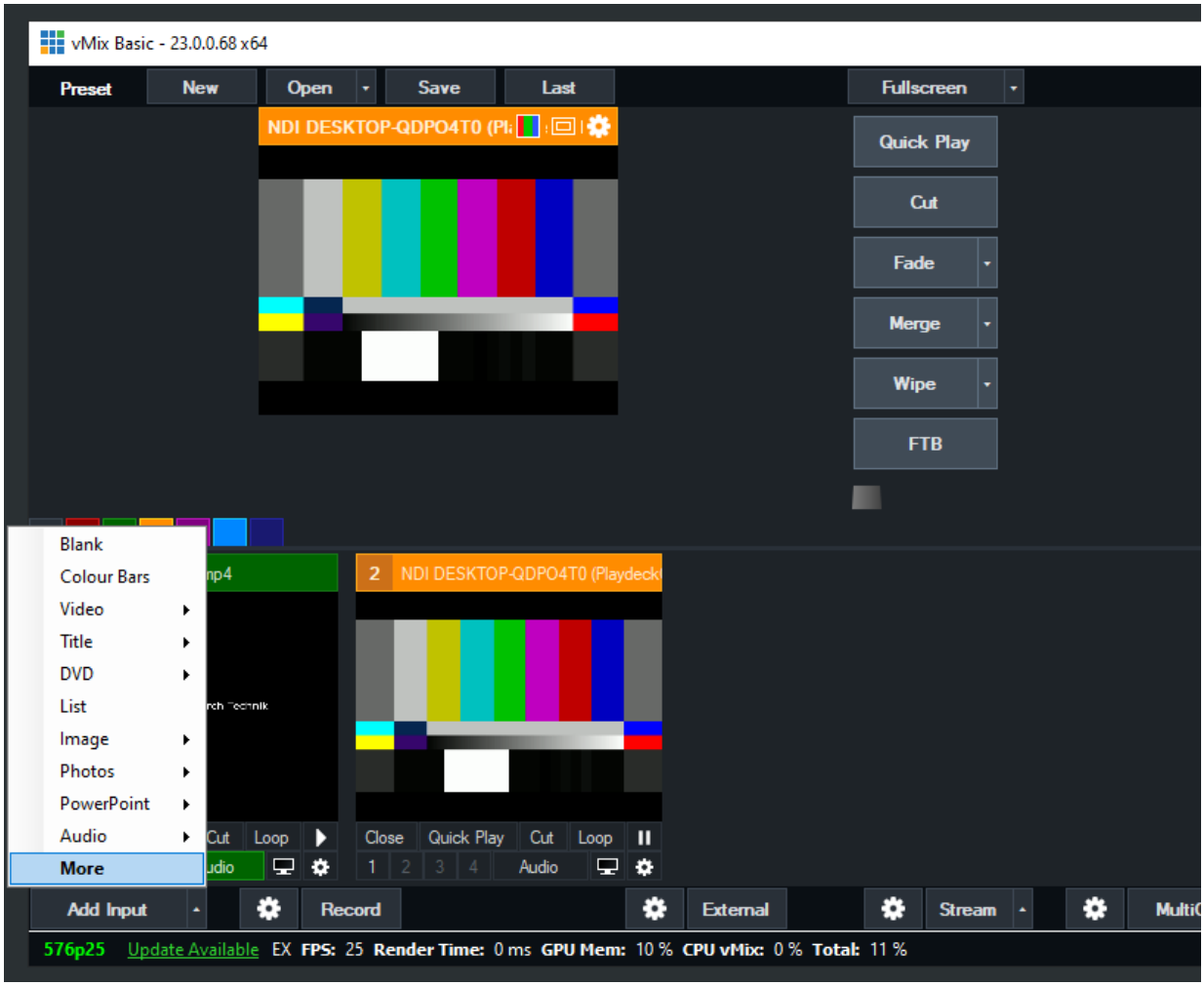
You only need to activate NDI as External Output like this:

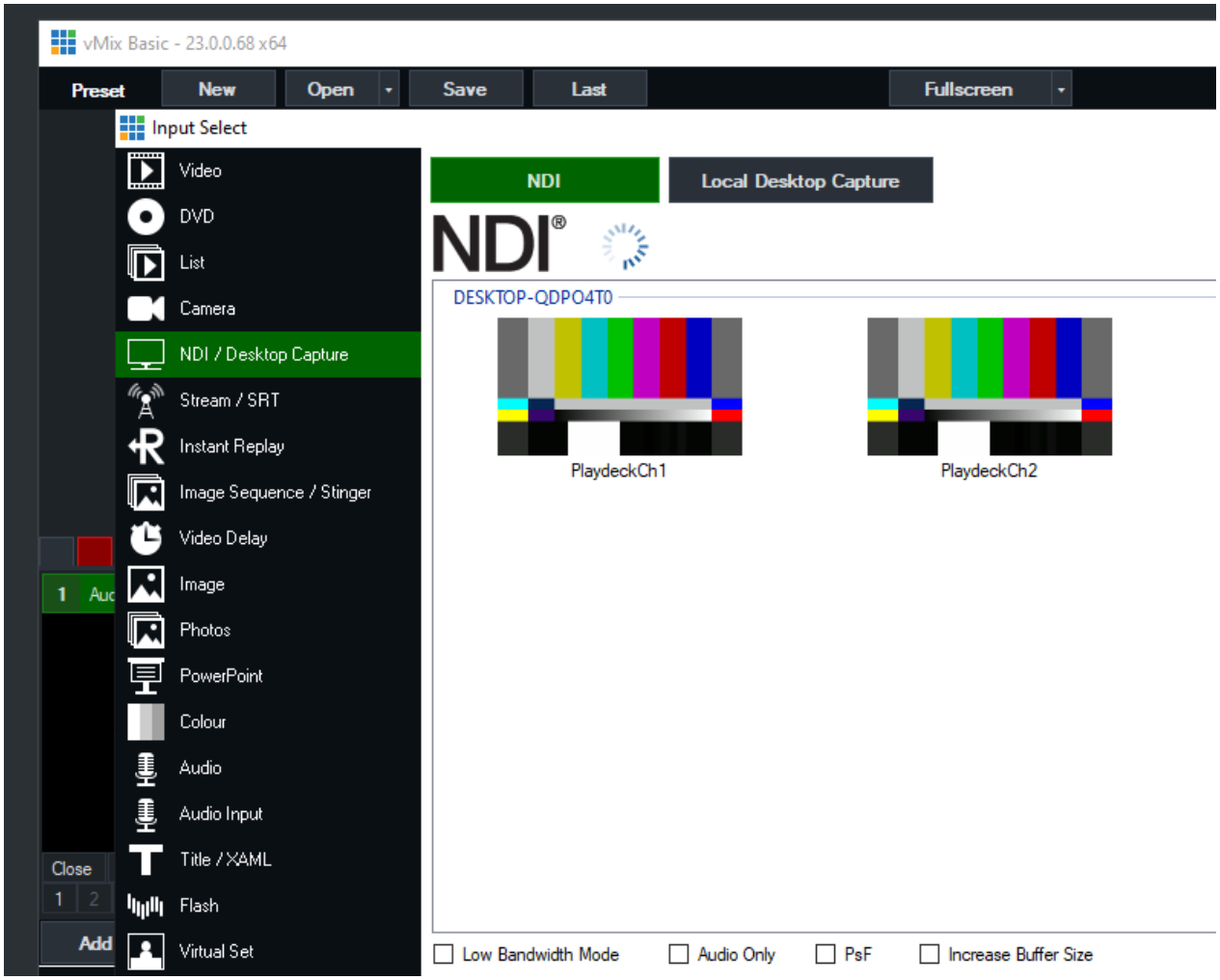




Send Video from PLAYDECK to vMix

Activate NDI Output in PLAYDECK and add the NDI as new Source in vMix like this:


vMix Basic - 23.0.0.68 x64
Preset
New
Open
Save
Last
Fullscreen
NDI DESKTOP-QDPO4T0 (Pl
Quick Play
Cut
Fade
Merge
Wipe
FTB
Blank
Colour Bars
Video
Title
DVD
List
Image
Photos
PowerPoint
Audio
More
2
NDI DESKTOP-QDPO4T0 (Playdeck
Cut
Loop
Close
Quick Play
Cut
Loop
1
2
3
4
Audio
Add Input
Record
External
Stream
576p25
Update Available
EX
FPS: 25
Render Time: 0 ms
GPU Mem: 10 %
CPU vMix: 0 %
Total: 11 %

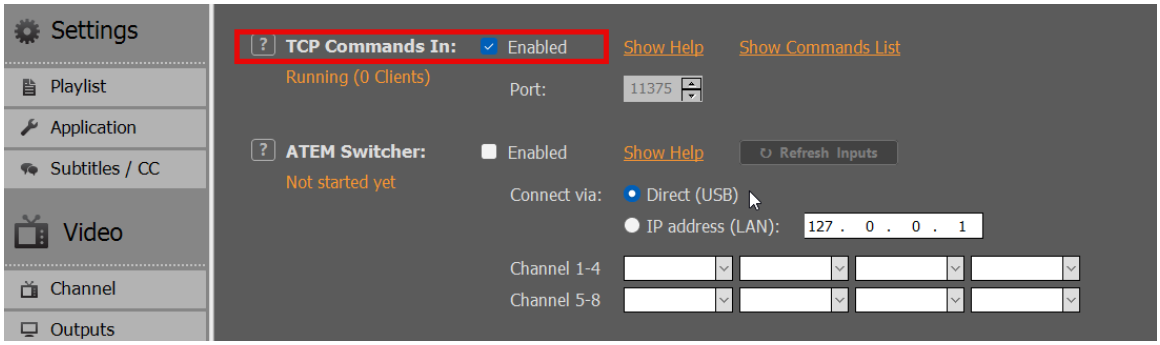

vMix Basic - 23.0.0.68 x64
Preset
New
Open
Save
Last
Fullscreen
Input Select
Video
DVD
List
Camera
NDI / Desktop Capture
Stream / SRT
Instant Replay
Image Sequence / Stinger
Video Delay
Image
Photos
PowerPoint
Colour
Audio
Audio Input
Title / XAML
Flash
Virtual Set
NDI
Local Desktop Capture
NDI®
DESKTOP-QDPO4T0
PlaydeckCh1
PlaydeckCh2
Low Bandwidth Mode
Audio Only
PsF
Increase Buffer Size
Close
Add

Using Companion for Playout Control

This article will show how to use bitfocus Companion together with PLAYDECK. Companion has a ready-to-use module for PLAYDECK. This is the most comfortable way to connect PLAYDECK to your ATEM Mixer or STREAMDECK.

Installation

Enable TCP Commands in PLAYDECK to allow 3rd party apps to control PLAYDECK externally:

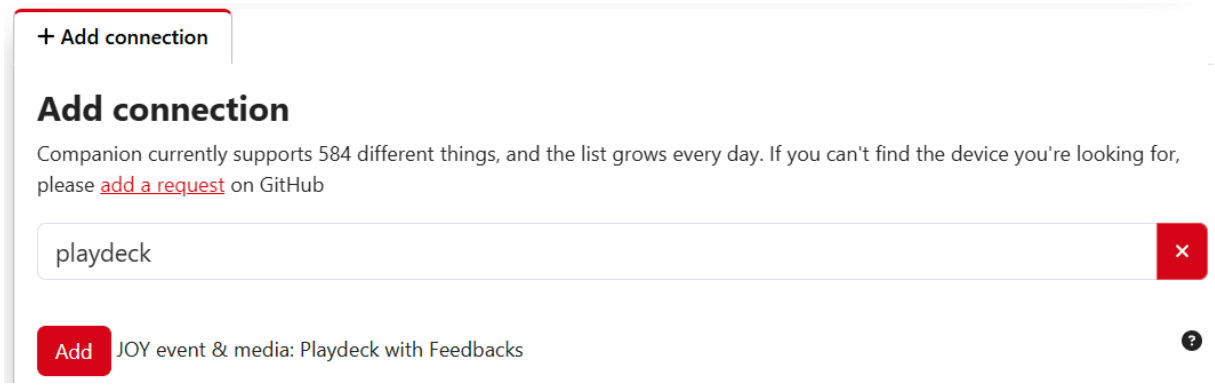


Download and install the latest Version of Companion. The PLAYDECK module is included in the installation:

https://bitfocus.io/companion

Note: The PLAYDECK Companion Module was developed by Semenov Nick. Visit his GitHub for the latest Module updates.

Start Companion, add a new connection and search for PLAYDECK to add the module:



Once connected, you can select typical PLAYDECK Actions to assign to Buttons, like CUE, PLAY, Start Overlay, and many more.

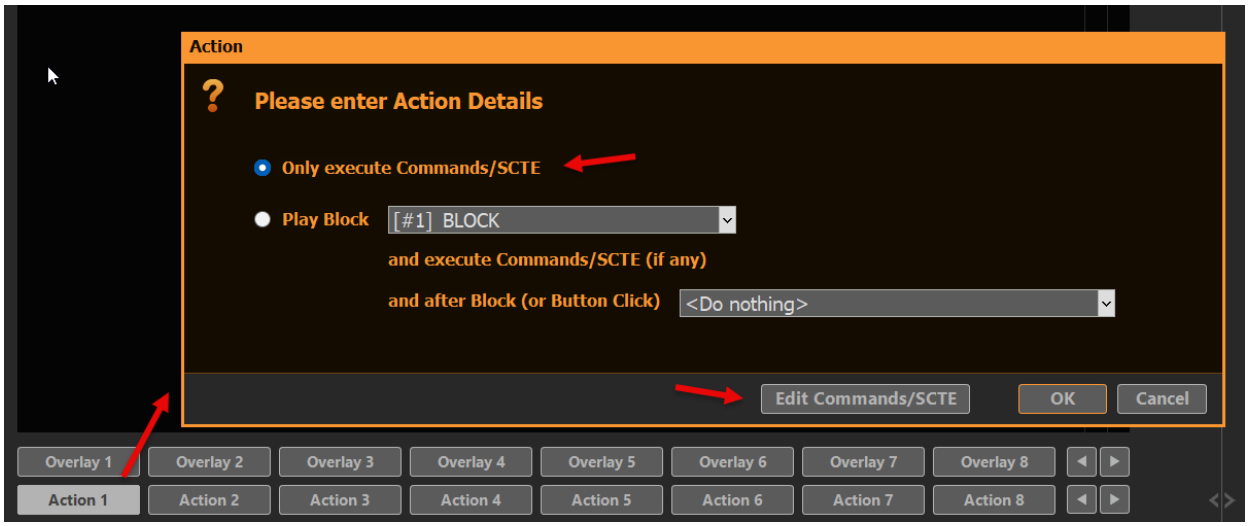
Using Commands for Playout Control

This article will show how to use Commands to control your Playout.

Example: CUE+PLAY multiple Channel

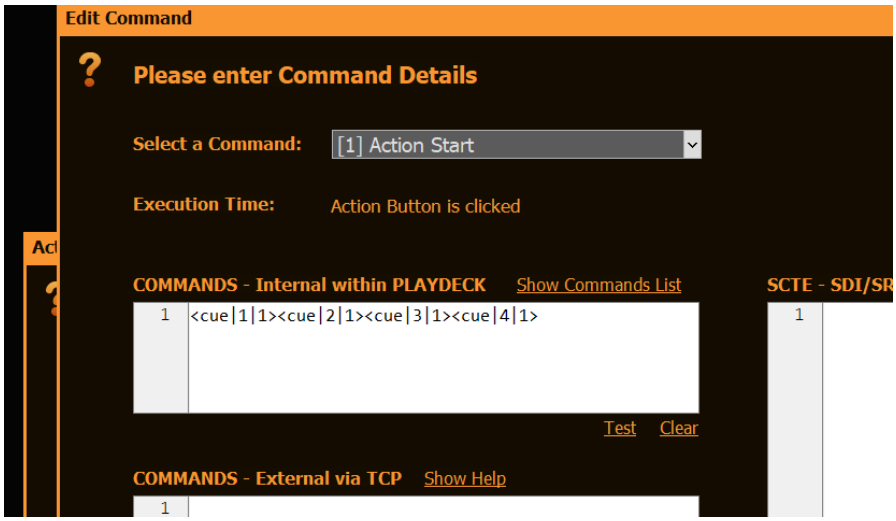
We want to create an Action Button, which will CUE several Output Channel and PLAY them simultaneously (almost synced).

Create a new Action Button by clicking on it and select COMMANDS, then EDIT COMMANDS:



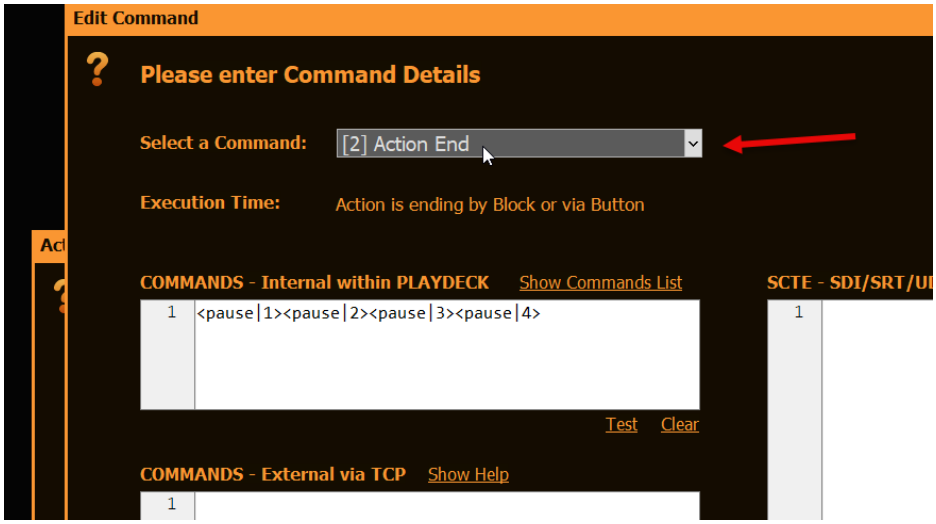
Add this Line for the first Click (ACTION START).
<cue|1|1><cue|2|1><cue|3|1><cue|4|1>

This will cue the first Clip if Channel 1-4:

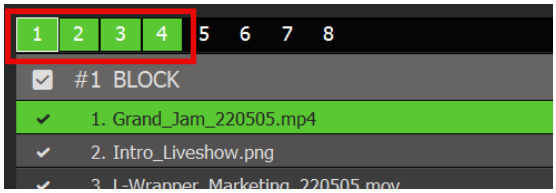


Add this Line for the second Click (ACTION END).
<pause|1><pause|2><pause|3><pause|4>

This will un-pause Channel 1-4 and send the CUE into PLAY:



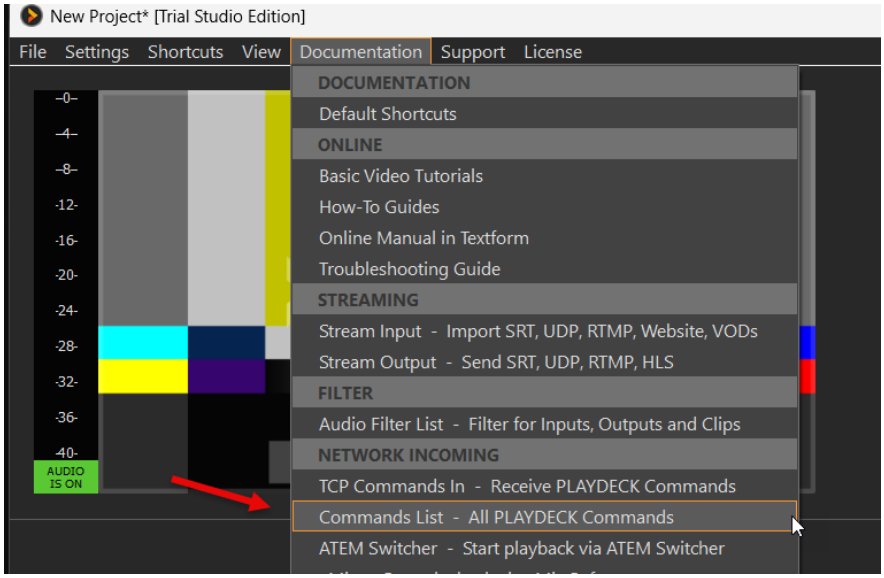
Lets test our new Button: Add a Clip to Channel 1-4 then press the Action Button. Not wait, until all Channel turn GREEN, indicating the Clip has been CUE'd:



Then press the Action Button again to start playback of all channel.

List of all Commands and where to add them

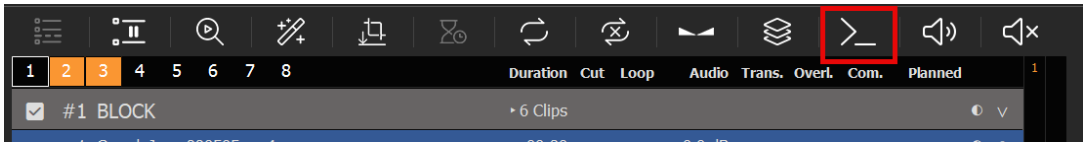
The List of all Commands can be quickly opened over PLAYDECK's Main Menu:



Commands can be added to:

- Action Buttons (Start, End)
- Overlay Button (Start, End)
- Clips (Any Time)
- Blocks (Start, End)

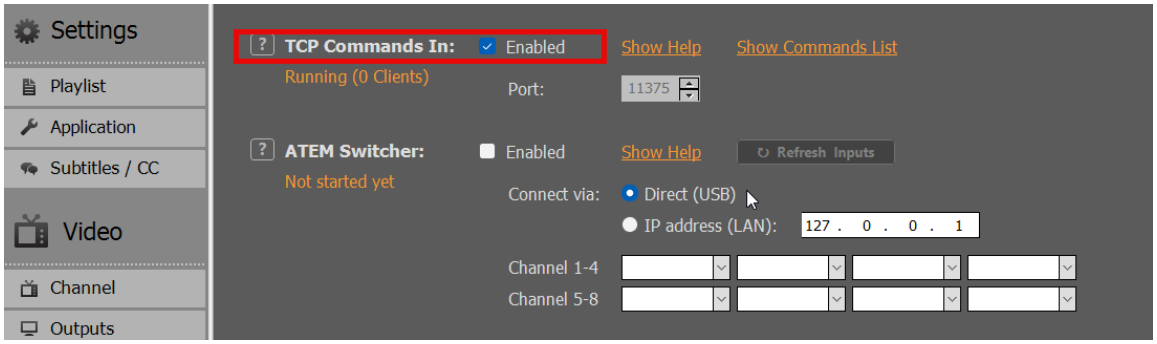
Mostly via this Icon:



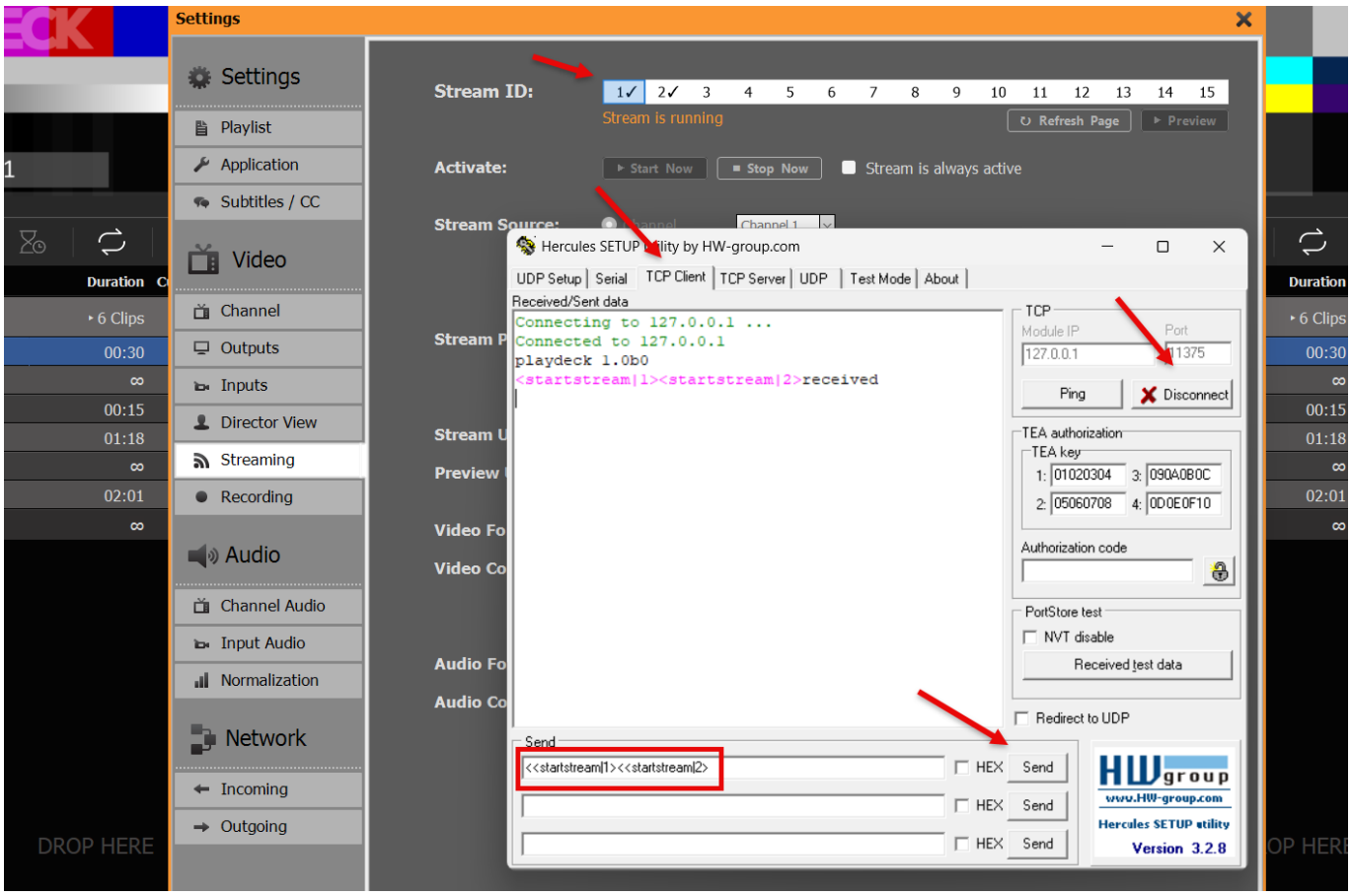
Send Commands from other Apps

We have a sparate article on how to control PLAYDECK with Companion.

But what, if it is not Companion, but a different 3rd party App? First, you would need to enable TCP Commands. This will open PLAYDECK up to receive and execute Commands from external sources:



In your 3rd party app, connect to PLAYDECK via TCP and send one or multiple Commands. In this Sample we use Hercules to send a Command, which will start Streams 1 and 2:

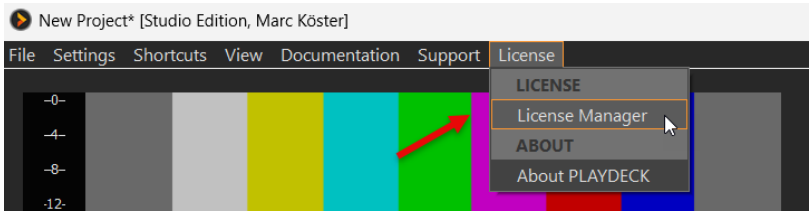


Move License to another PC

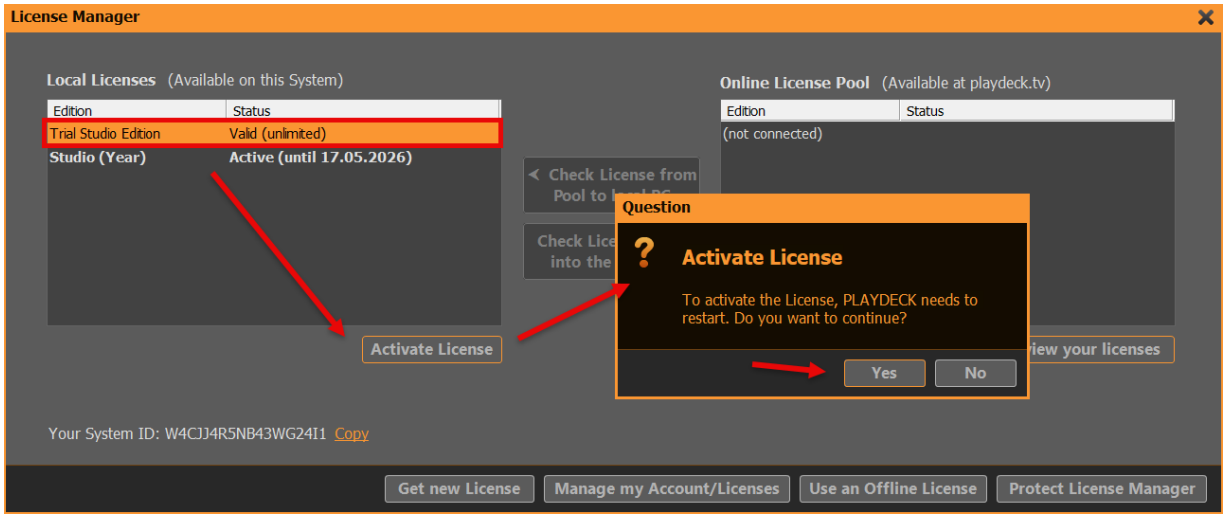
This article will show how to move your License from one System to another System.

Shop License

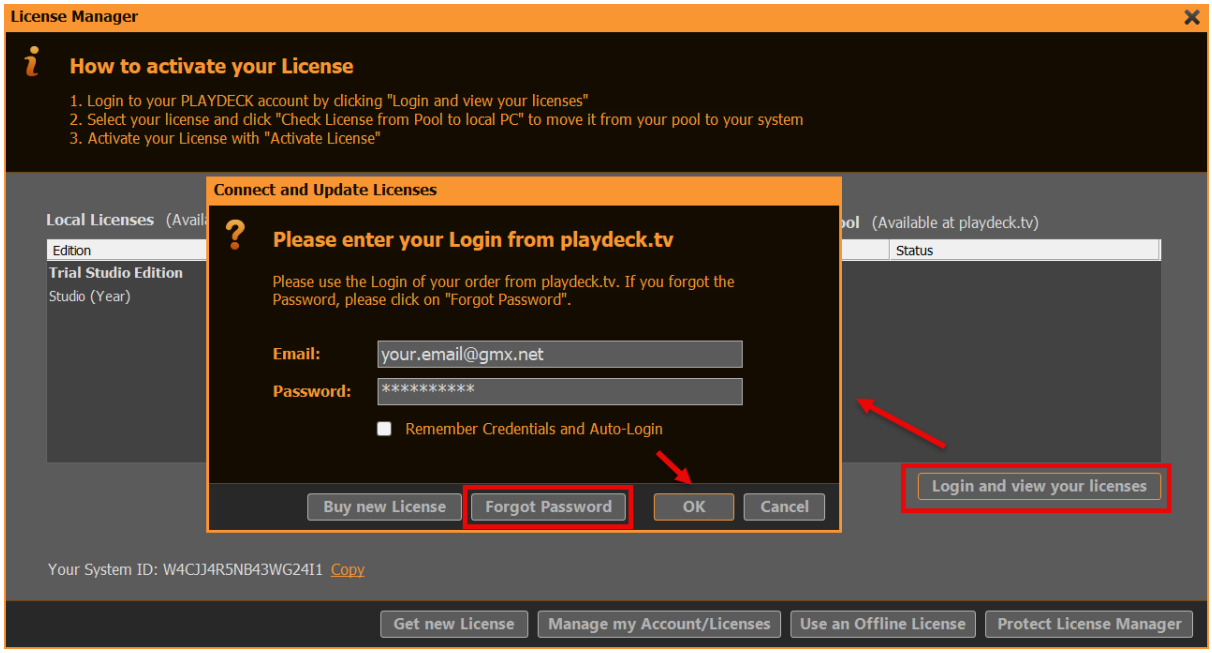
You can move your License via a simple Check-In/Out System. On the System with the active License, go into your License Manager:



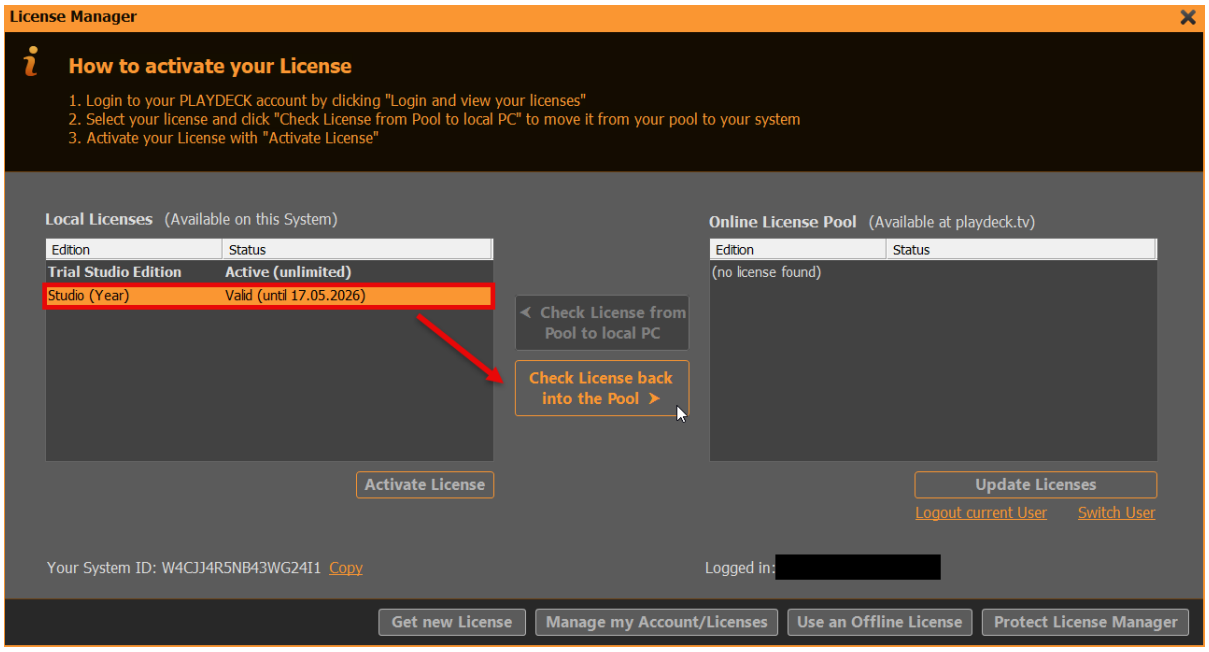
First, we want to free the active License, so we can move it. Select the TRIAL Edition and click ACTIVATE, then confirm the restart query:



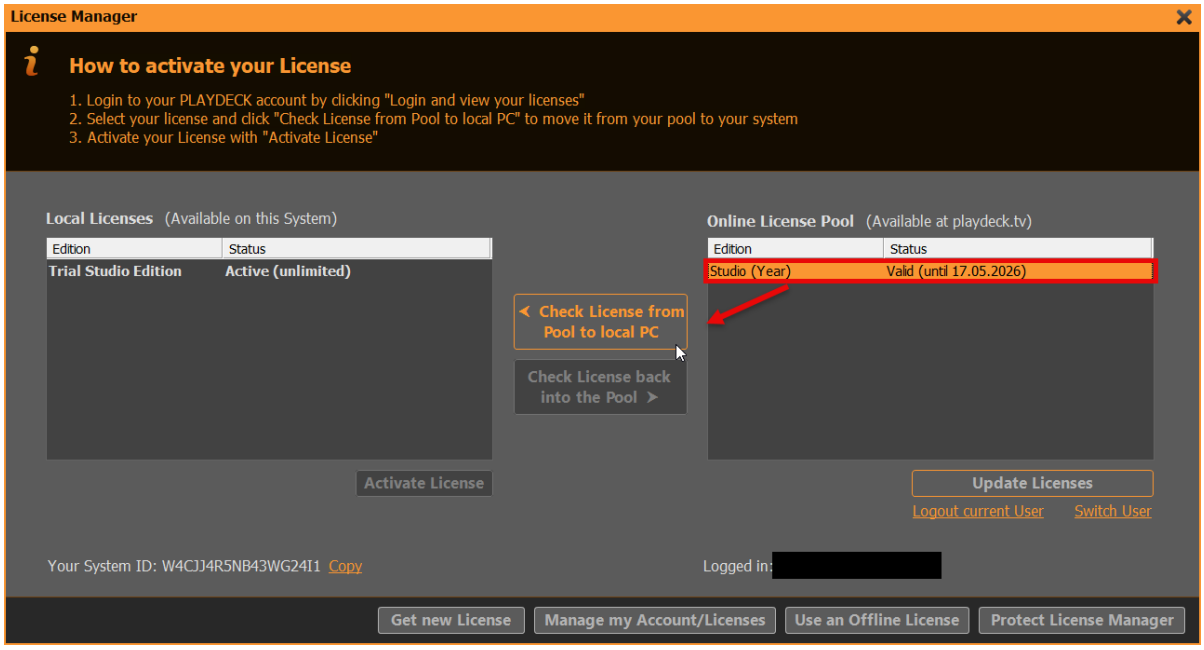
After restarting PLAYDECK, we can now move the free'd License back to your Online License Pool. Re-open the License Manager. Now login to your PLAYDECK ACCOUNT. Use the email adress and password, which you used during CHECKOUT in the SHOP. Use the FORGOT PASSWORD button, if you need to reset your password:



We can now move the License from you LEFT Side to the RIGHT Side with a Button Click (CHECK LICENSE BACK INTO THE POOL). This means, that the License becomes UNAVAILABLE on the current System and becomes AVAILABLE on ANY other System:



The next Screen shows the OTHER PC after logging into the License Manager. You simply reverse the Process now: Select the License on the RIGHT Side and move it with the Button to the Left Side (CHECK LICENSE FROM POOL TO LOCAL PC). PLAYDECK will now offer to activate the License and restart:



Dongle License

Simply detach the USB Dongle and attach it to another PC, then start PLAYDECK on that PC. Please note, that if you detach the Dongle while PLAYDECK is still running, PLAYDECK will close immediatly and without prior notice. It is recommended to close PLAYDECK regulary to ensure, that all Data is saved properly.

Offline License

The Offline License is bound to the PLAYDECK System ID and can NOT be moved to another System by the Customer. Please write us at support@playdeck.tv to receive a new Offline License.

Backup License

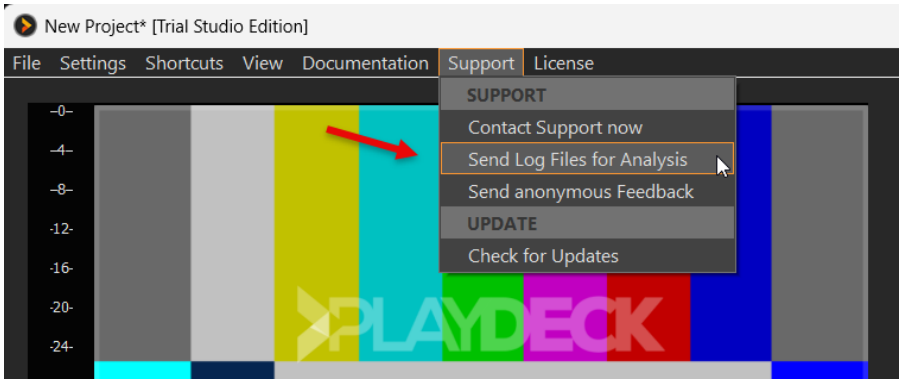
We have a separate article on how to use the Backup License.

Send Log Files to Support Team

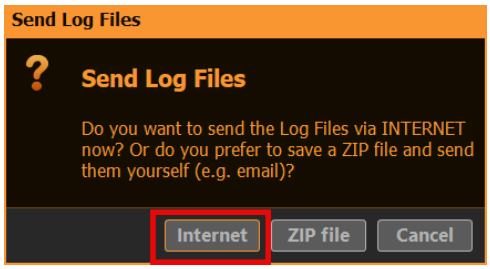
This article will show how to send you PLAYDECK Log Files to our Support Team.

Send via Internet

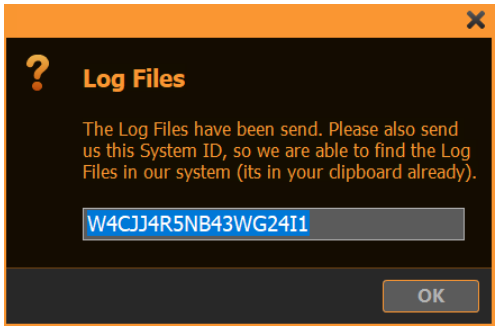
If you have an active Internet Connection on your PLAYDECK System, open the Main Menu and select SUPPORT and then SEND LOG FILES:



Then select the INTERNET Option:

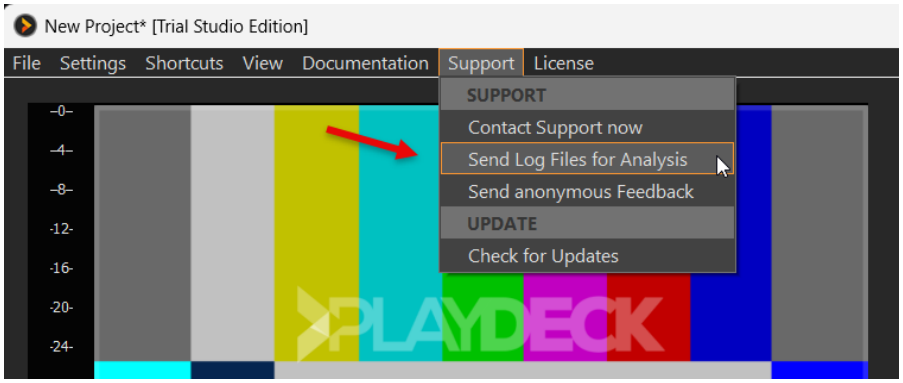


After the upload is complete, please send us a message to support@playdeck.tv along with your System ID. If you seen the following Popup, the System ID is in your Clipboard already and you can simply paste it with CTRL+V into your email message:

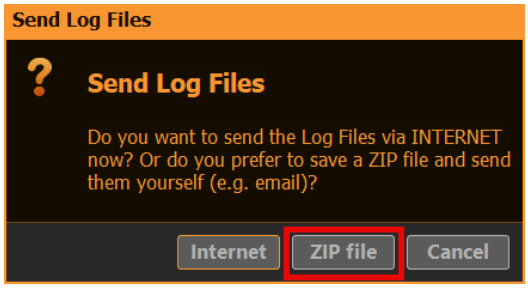


Send offline via ZIP File

If you DONT have an active Internet Connection on your PLAYDECK System, open the Main Menu and select SUPPORT and then SEND LOG FILES:



Then select the ZIP FILE Option:



After you pick a destination Folder on your System, a 7-ZIP-file will be written there:

Please send us a message to support@playdeck.tv along with your ZIP-file.

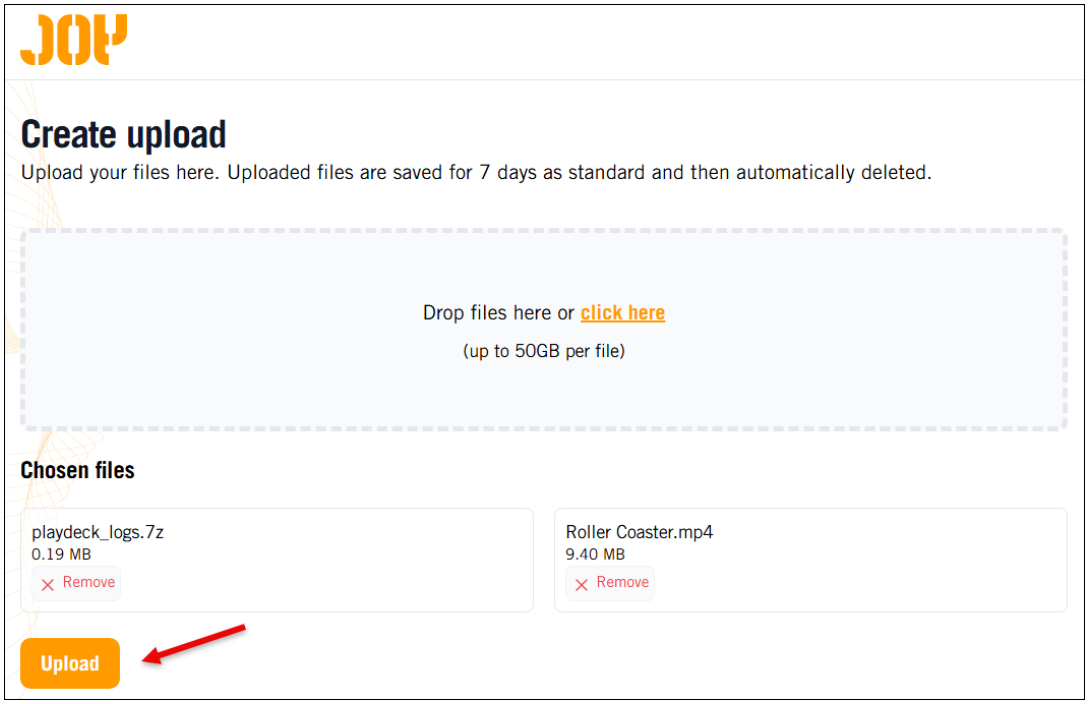
If your Email-system doesnt allow sending Files or the ZIP-file itself is too large (> 5 MB), you can also use our Downloadportal to upload the File.

Upload Files to Support Team

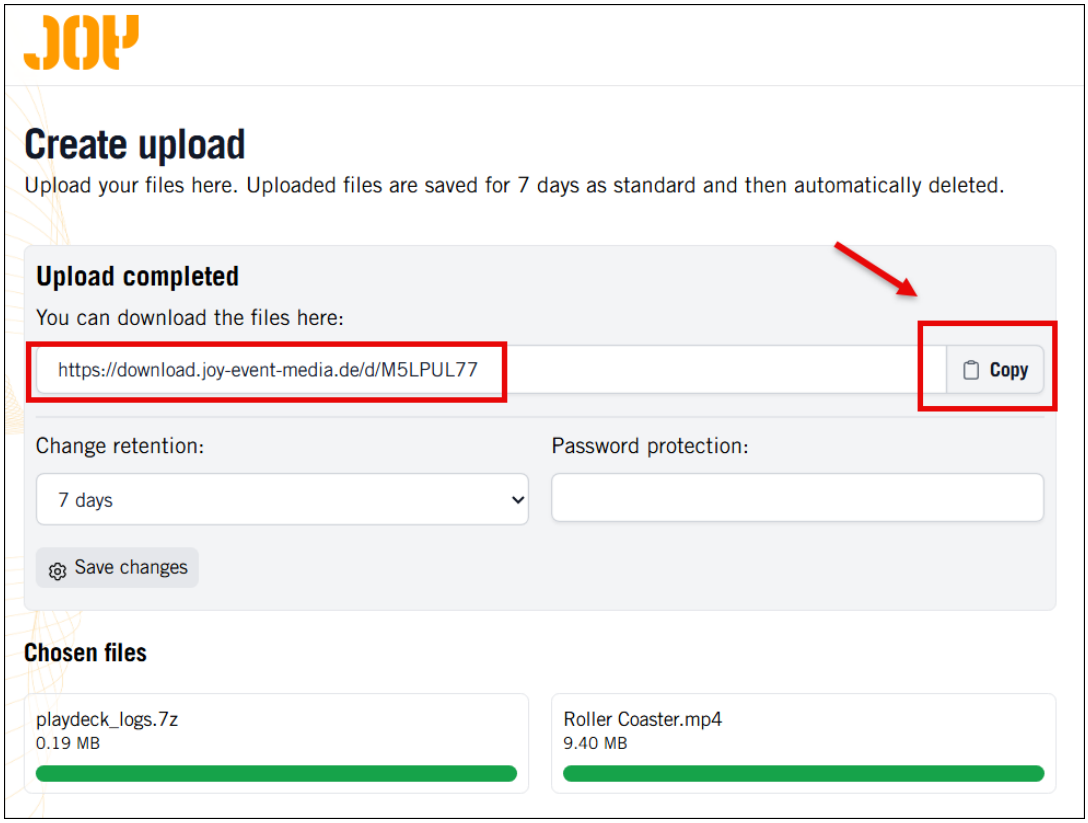
This article will show how to upload Files and send them to the PLAYDECK Team.

Open our Downloadportal via this URL: https://download.joy-event-media.de/

You can add your Files now via Drag-drop or file-browser. In our example we already added some PLAYDECK Log-files and a sample video. If you are done adding, please click UPLOAD.



After the Upload is complete, please click COPY to put the Download-link into the Clipboard. Then write us a message to support@playdeck.tv along with that Download-link:



Note: Without the Download-link we cant access your Files. This is to protect Customer data.

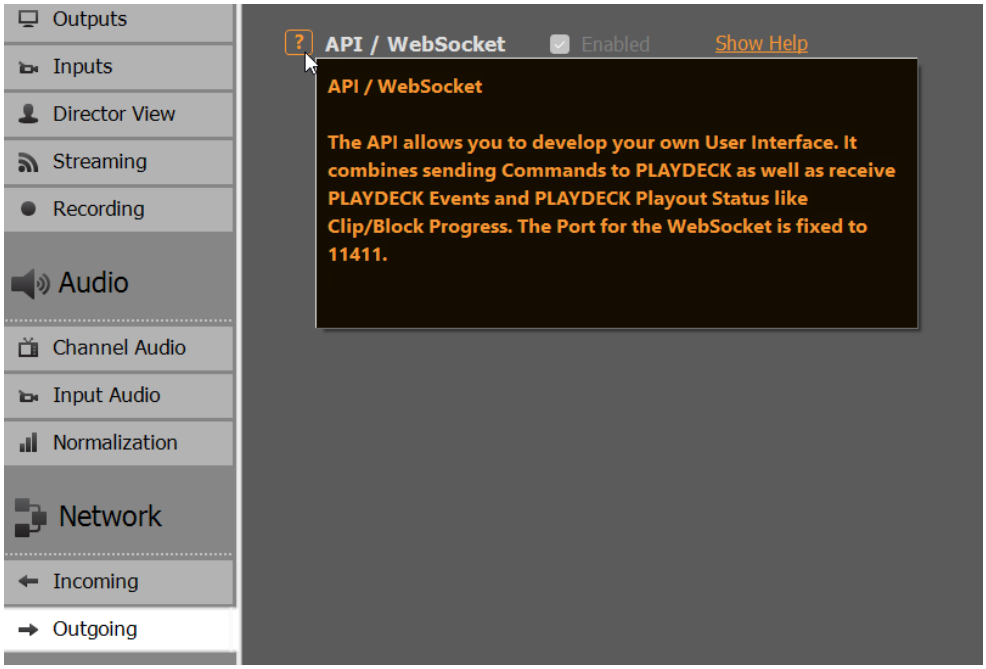
Free Backup License

This article will show how to use the Free Backup License, which comes with every PLAYDECK License. The Backup License functions as follows:

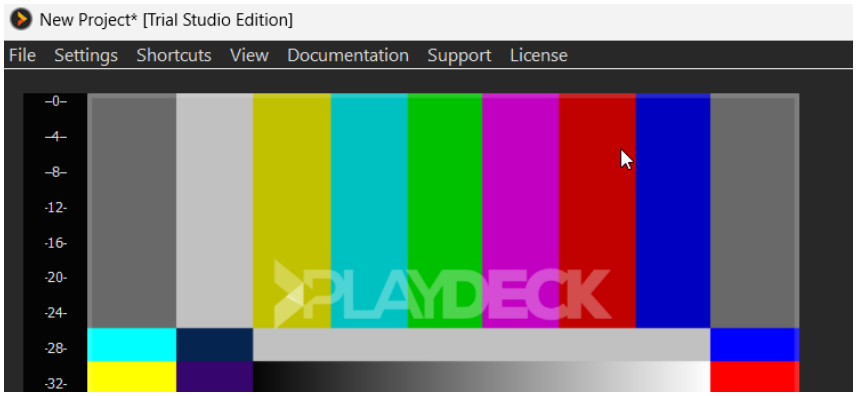
- Can be used with ANY PLAYDECK License (Dongle, Shop, Offline)
- Can be used on ANY PLAYDECK-PC in the same network
- Multiple Clients are supported
- 24/7 operation is NOT supported (max 24h per session)

Setup

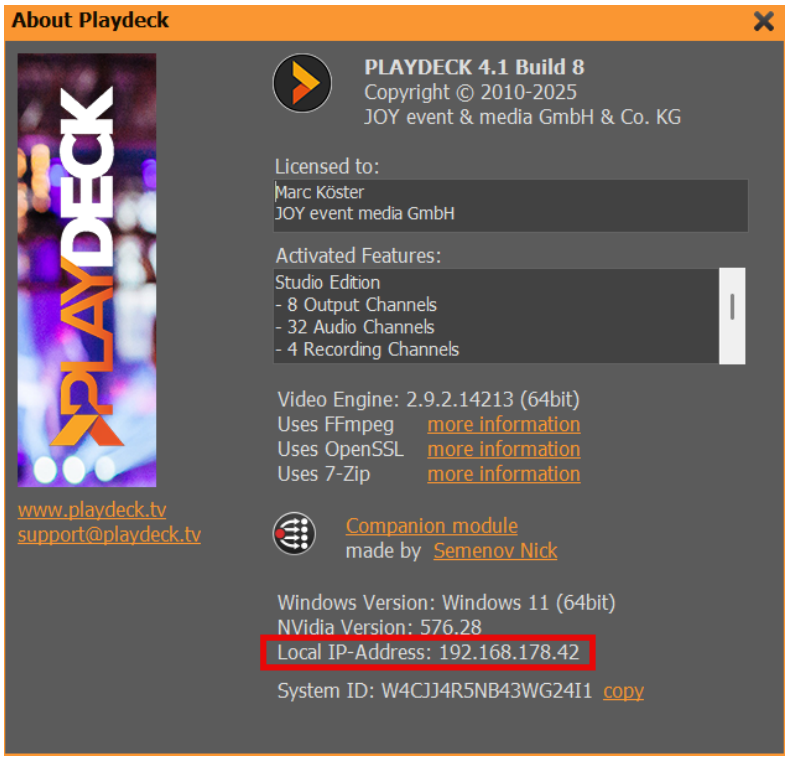
On your MAIN System (the one WITH the License), you dont have to do anything: The BACKUP LICENSE SYSTEM is enabled by default. It will use the WebSocket Port 11411, so make sure this Port is not blocked:



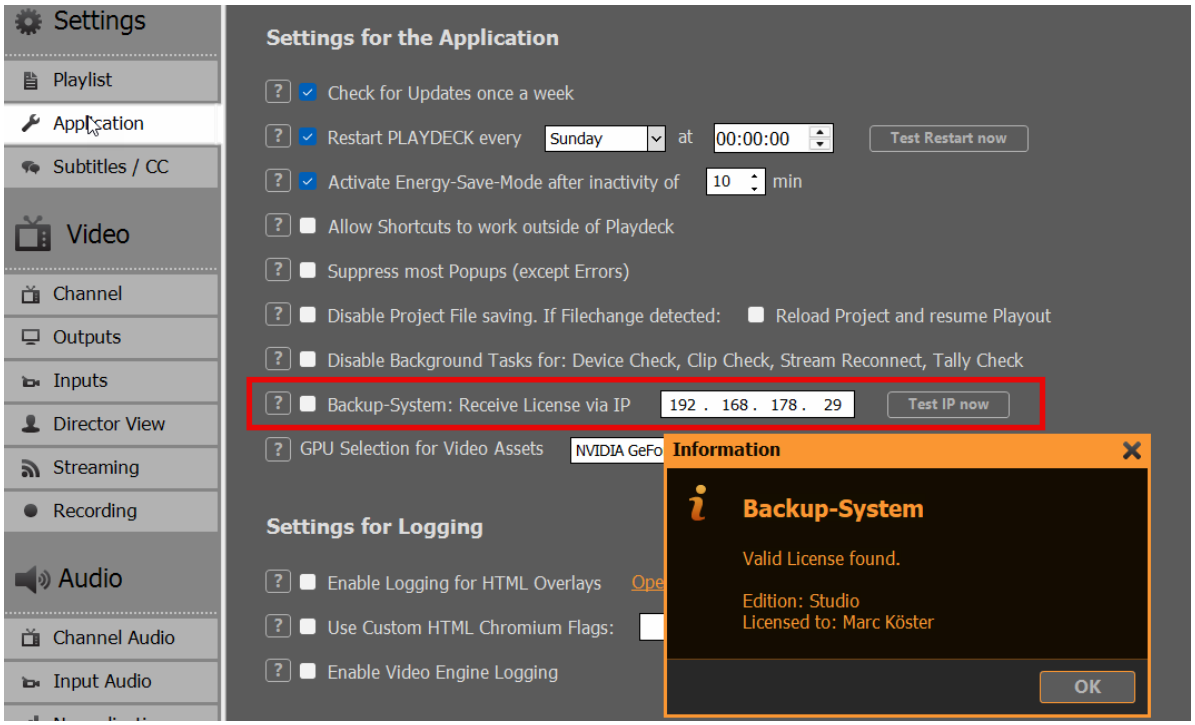
On the CLIENT System (the one WITHOUT the License), you perform a regular PLAYDECK installation, nothing special here: PLAYDECK will not start in TRIAL MODE with the watermark:



Now we need to tell CLIENT PLAYDECK, where to find the MAIN PLAYDECK, by providing the IP adress of MAIN. To receive the IP adress of your MAIN, you can simply open the ABOUT BOX in PLAYDECK (Main Menu > License > About):

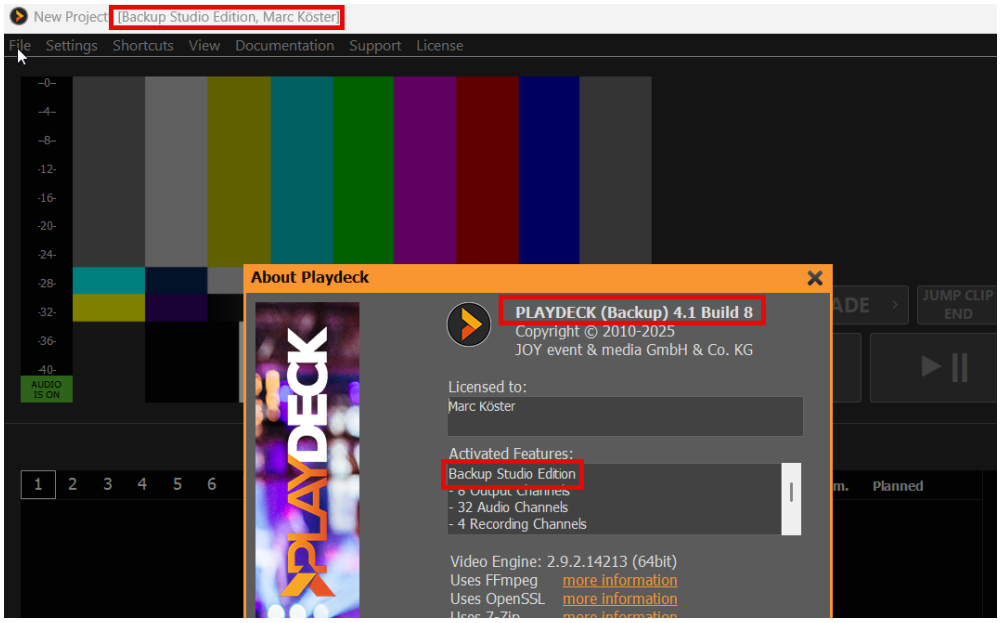


On your CLIENT, go to Application Settings, enter the IP adress of MAIN PLAYDECK and click TEST IP NOW. The CLIENT now tests the Network connection and searches for a valid License on MAIN. If you dont see the following INFO POPUP, please check your MAIN License and your Network connection between MAIN and CLIENT:



You can now enable the feature by clicking BACKUP-SYSTEM. PLAYDECK will restart now with the exact same License of MAIN. You can verify, the CLIENT

is using the Backup License correctly in several Displays:



Note: If the CLIENT has a valid License itself, the License of MAIN will NOT be used.

Rules of Backup-License

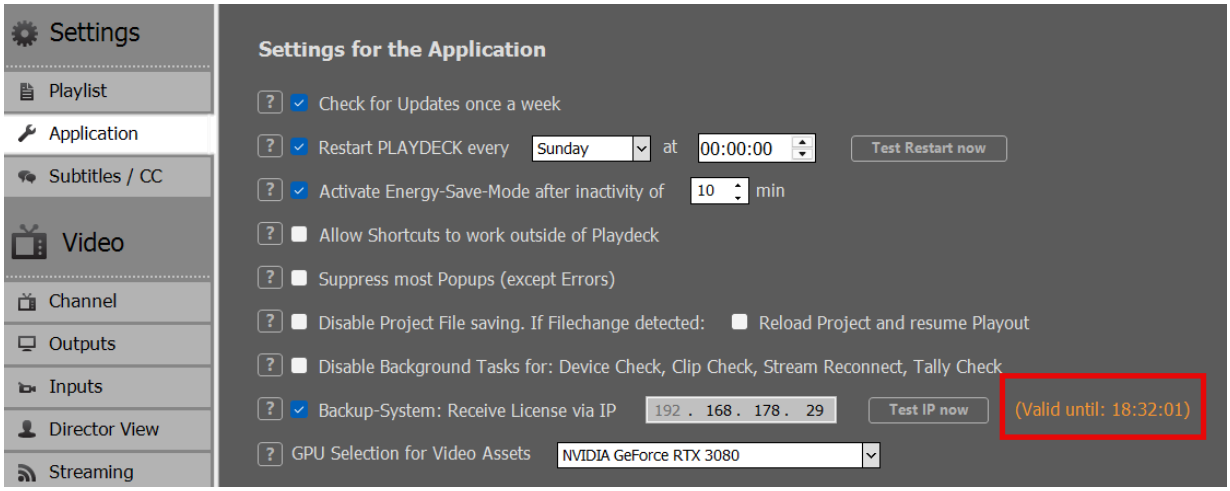
The Backup-License is not meant for 24/7 operation, but to provide a fail-safe for immediate operation, e.g. in a live production, and without the need to buy a secondary license, just to have a backup system for some use cases.

For unattended 24/7 operation, you would need to purchase a secondary license. Please contact use at support@playdeck.tv, if your next License is for Backup only, and we will provide you with a appropriate discount.

The MAXIMUM RUNTIME of your BACKUP LICENSE will be:
– 2 hours, if the MAIN gets disconnected or closed
– 24 hours, if then MAIN stays online

PALYDECK will terminate the CLIENT without prior warning. So please make sure, to bring the MAIN back online within the 2 hour window, or to restart both MAIN and CLIENT within the 24 hour window.

You can control the remaining RUNTIME in the Settings of the CLIENT:

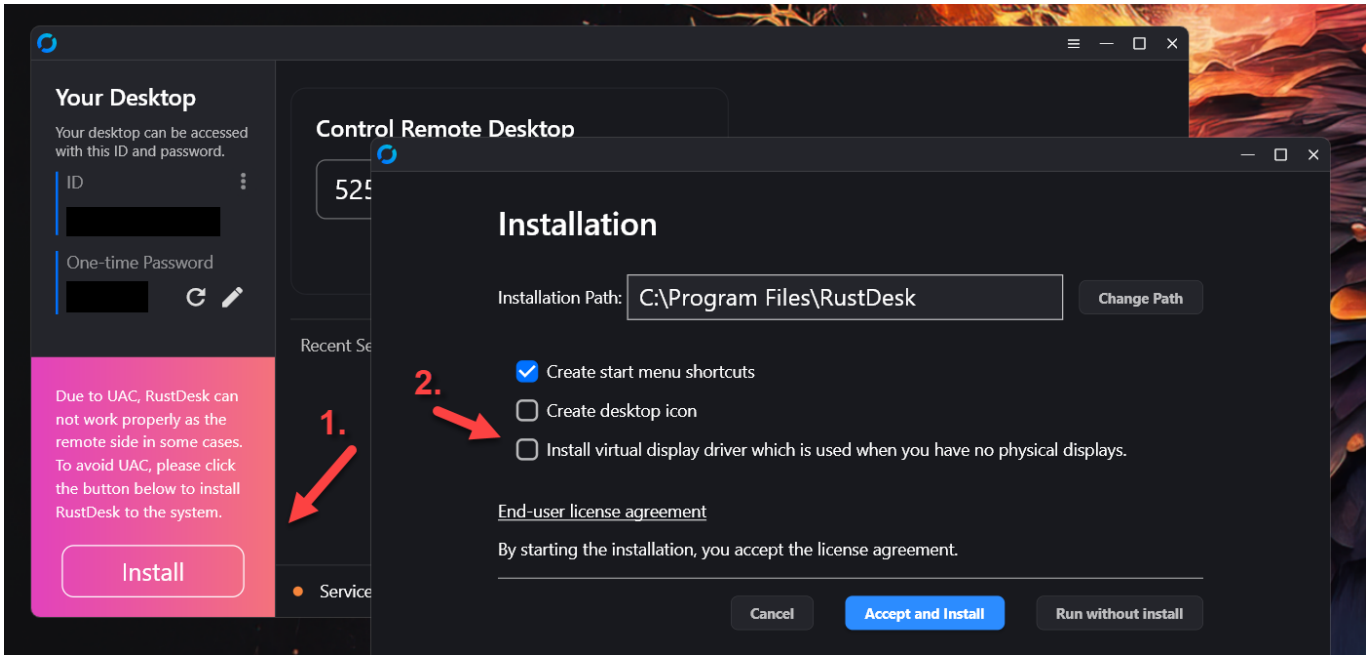

Settings
Playlist
Application
Subtitles / CC
Video
Channel
Outputs
Inputs
Director View
Streaming
Settings for the Application
Check for Updates once a week
Restart PLAYDECK every
Sunday
at
00:00:00
Test Restart now
Activate Energy-Save-Mode after inactivity of
10
min
Allow Shortcuts to work outside of Playdeck
Suppress most Popups (except Errors)
Disable Project File saving. If Filechange detected:
Reload Project and resume Playout
Disable Background Tasks for: Device Check, Clip Check, Stream Reconnect, Tally Check
Backup-System: Receive License via IP
192 . 168 . 178 . 29
Test IP now
(Valid until: 18:32:01)
GPU Selection for Video Assets
NVIDIA GeForce RTX 3080

Setup RustDesk unattended Access for Remote Support

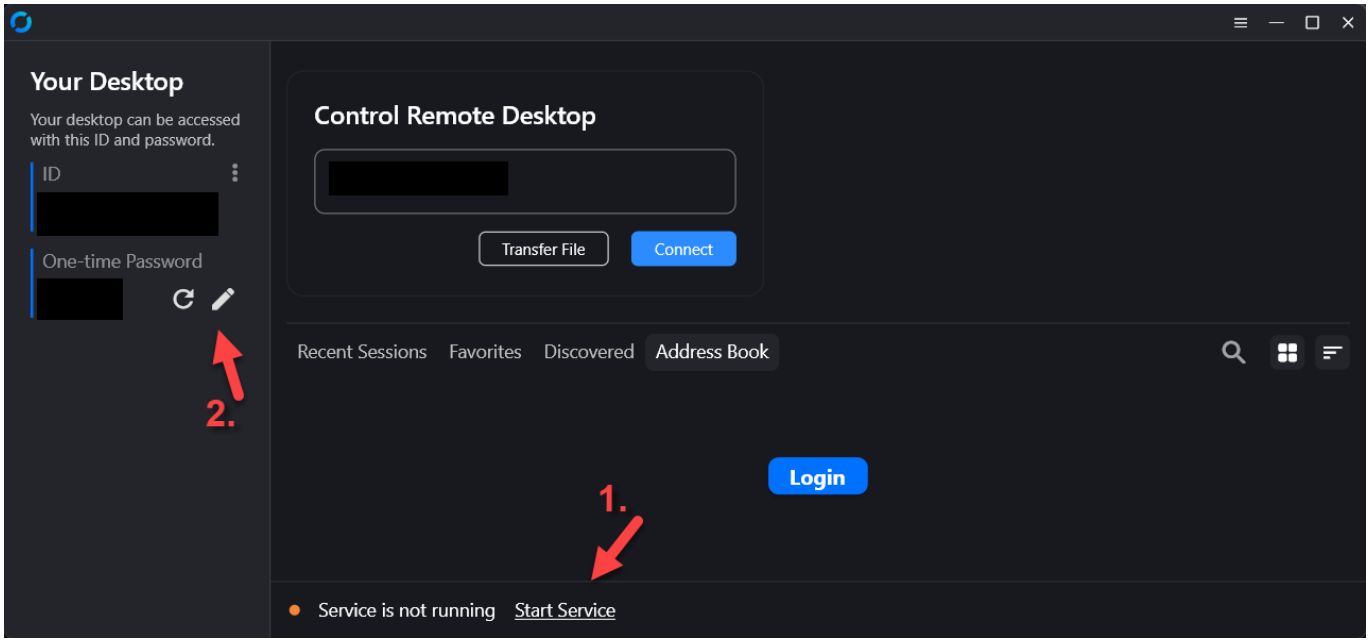
RustDesk is a free Remote Destop Software, which we use to login to remote systems. The main advantage of RustDesk over other Remote Desktop Connections is, that is does interfere with installed GPUs (eg. installing a Virtual GPU), therefore not interfering with PLAYDECK during Remote Access.

1. To get started, download our customized RustDesk from our website. This version will use our private RustDesk server (and not the public server) to protect your data and improve the connection speed.
https://get.remote-joy-event-media.de/

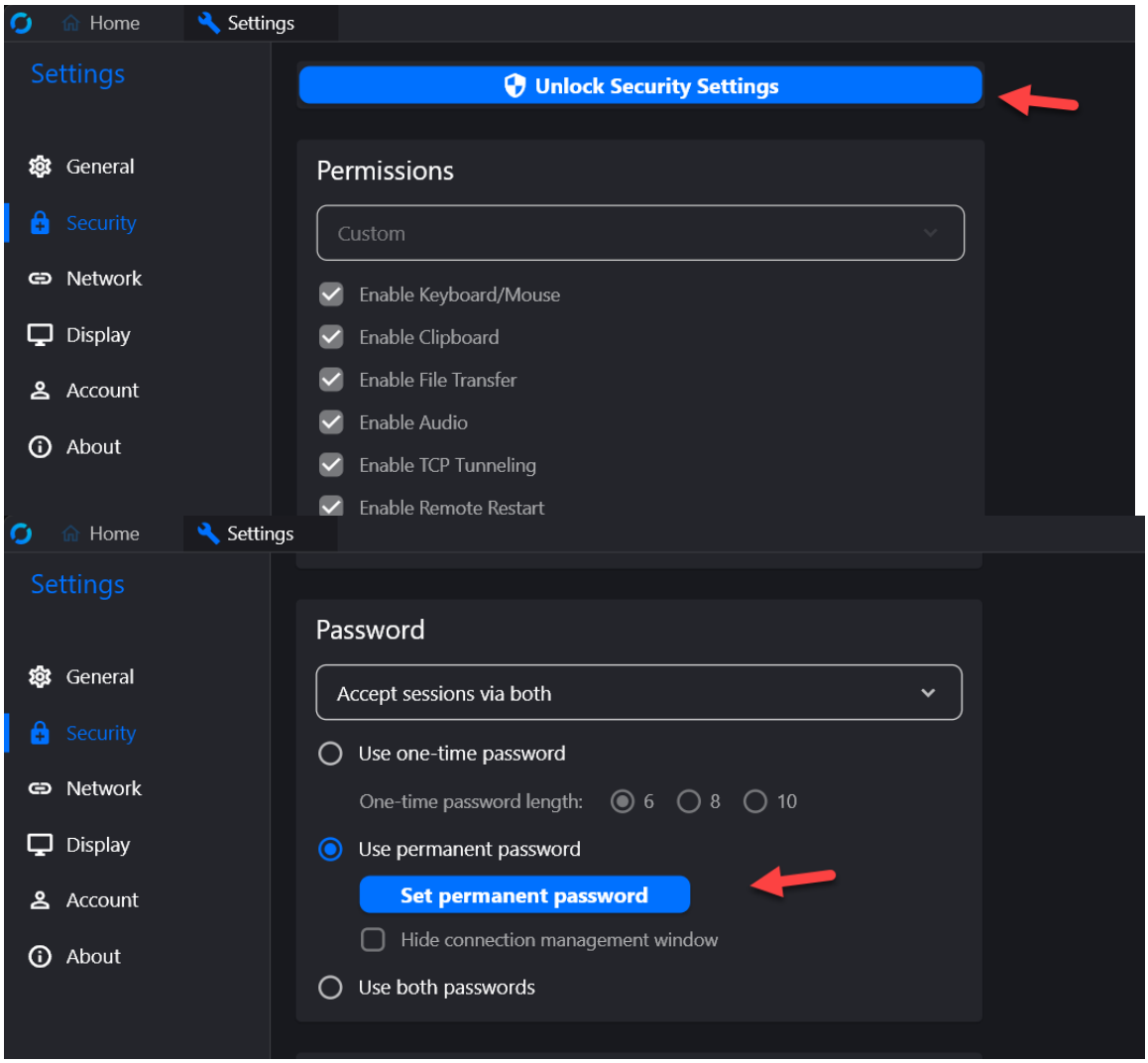
2. Once you start RustDesk for the first time, you will need to install it. Please deactivate “Install virtual display driver”, which might interfere with PLAYDECK.



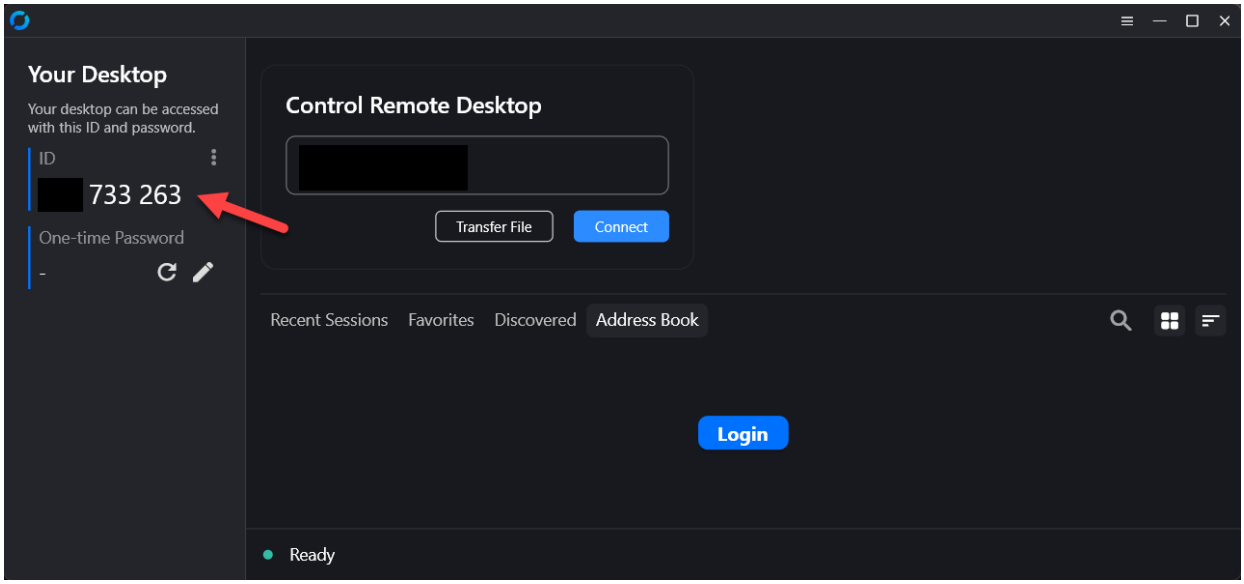
3. After re-starting RustDesk, you will not need to activate “Start Service”. This is important to elevate priviliges to e.g. open the device manager. After that you need to setup a permanent password: Click on the Edit Icon next to “One-time Password”.



4. Then click on “Unlock Security Settings” and scroll down.



5. Please send us your ID together with your permanent password to support@playdeck.tv



6. (Optional) If possible, please start RustDesk on a secondary PC (No need to install) and test the connection to the System you have setup for remote support. Now thru this remote connection, on the secondary system, please try to:

– Open Device Manager
– Open Task Manager
– Open NVidia Panel
– Copy any File to this Folder: c:\Program Files (x86)\JoyEventMedia\Playdeck\

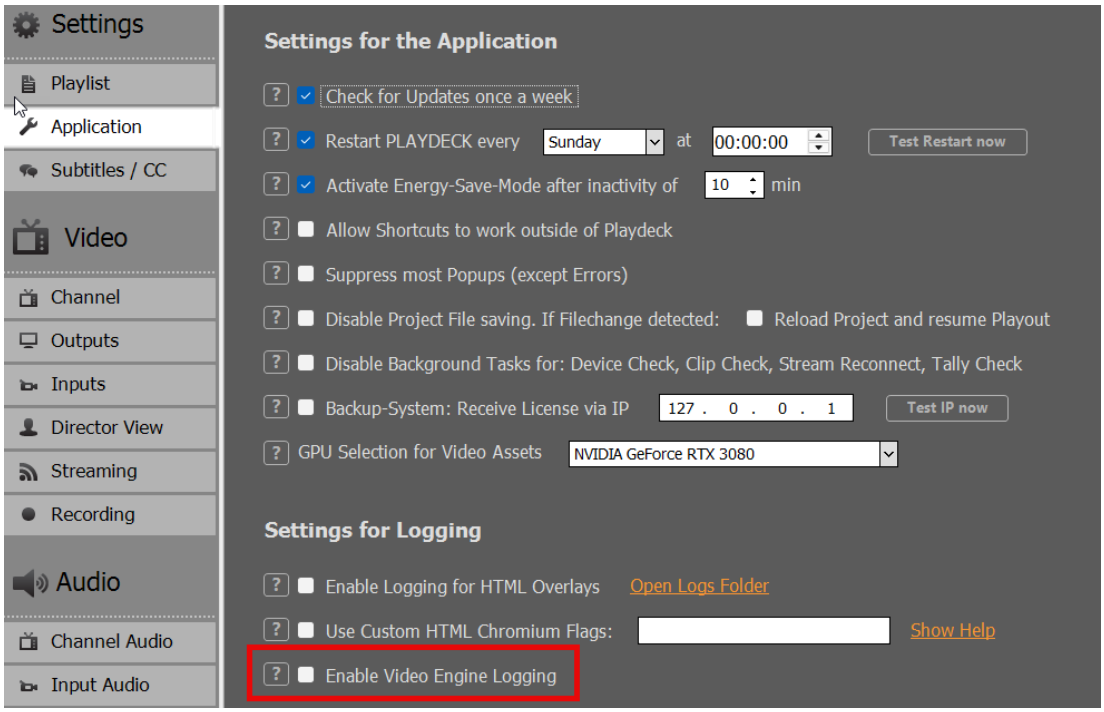
These tests make sure, that all needed support actions can be done remotely. If is likely, that any Anti-Malware or other Protection software breaks the connection. In that case please de-activate those Tools temporarily for the remote support session.

Video Engine Logging

This article will show how to enable detailed Video Engine Logs for us to analyze.

1. Enable Logging

In PLAYDECK goto Application Settings and activate “Enable Video Engine Logging”. PLAYDECK will now ask you to restart.



2. Reproduction

After restarting now please reproduce the Issue/Problem at hand. Please make a note of the current Date/Time (and let us know). After the Issue/Problem has occured, close PLAYDECK (or use Task Manager to terminate). Do NOT OPEN PLAYDECK again, otherwise the Logs will be lost, as they are being cleaned on every Start to reduce Upload Size.

Note: The shorter the Logs the better, as it is more easy to isolate the issue. Make sure to start PLAYDECK only to reproduce the issue and don’t let PLAYDECK run for a long time.

3. ZIP’ing and sending the Logs

Create a new ZIP File and add the following Folders:

*c:\ProgramData\JoyEventMedia\Playdeck**crashs***
*c:\ProgramData\JoyEventMedia\Playdeck**logs***
*c:\ProgramData\JoyEventMedia\Playdeck**sdklogs***

Note: The ProgramData Folder is hidden in Windows. You have to enter it manually into your File Explorer.

Note: To create a ZIP File in Windows, mark the above Folders with CTRL, then right-click on any marked Folder and select “Compress to ZIP File”.

Upload the ZIP to our Download Server:
https://download.joy-event-media.de/

After uploading click the COPY Button and send the Link to support@playdeck.tv.
Please dont forget to send the Date/Time of the Issue/Problem occurance, so we can easily find it in the Logs.

4. Clean Up

You can now safely delete your ZIP File.

Also make sure to deactive the Setting “Enable Video Engine Logging” in PLAYDECK, as it will slow down your PLAYDECK performance. This setting is not meant to be enabled all the time.