

# I<sup>U</sup>VERSAL

---

## *Vertex User Manual*

---

1.	<i>What is Vertex?</i>	8
2.	<i>Welcome</i>	10
3.	<i>Introduction</i>	13
3.1	<i>Licenses and Editions</i> .....	15
3.2	<i>User Interface and Fullscreen Renderer</i> .....	19
3.3	<i>Multi-System Application</i> .....	20
4.	<i>Installation and Setup</i>	21
4.1	<i>Hardware Requirements</i> .....	23
4.2	<i>Installation</i> .....	26
4.3	<i>Trial Version</i> .....	32
4.4	<i>License Activation and Deactivation</i> .....	33
4.5	<i>FAQ Performance Recommendations</i> .....	57
4.6	<i>Release-Notes</i> .....	62
5.	<i>Getting Started</i>	64
5.1	<i>Canvas, Surface and Output</i> .....	67
5.2	<i>Configure Outputs</i> .....	69
5.2.1	<i>Surface</i> .....	69
5.2.2	<i>Output</i> .....	80
5.2.3	<i>Warping</i> .....	84
5.2.4	<i>Softedge Blending</i> .....	88
5.2.5	<i>Audio Outputs</i> .....	96
5.3	<i>Content Editing, Composition &amp; Playout</i> .....	102
5.3.1	<i>Canvas</i> .....	104

5.3.2	Playback .....	106
5.3.3	Sequence .....	113
5.3.4	Track .....	122
5.3.5	Clip Container .....	128
5.3.6	Keyframes .....	145
5.3.7	Cue .....	149
5.3.8	Playlist .....	152
5.3.9	Clip .....	166
5.3.10	Geometry Modifiers .....	167
5.3.11	Video FX .....	168
5.3.12	Audio .....	169
5.3.13	Devices .....	173
5.3.13.1	DMX Input Device .....	181
5.3.13.2	DMX Output Devices .....	185
5.3.13.3	Powerpoint Remote .....	193
5.3.13.4	Stream Deck .....	200
5.3.13.5	Vertex NDI Streamer .....	206
5.4	Content Types .....	213
5.4.1	Import Content .....	214
5.4.2	Video .....	223
5.4.3	Image Sequence .....	233
5.4.4	3D Objects .....	238
5.4.5	Images .....	238
5.4.6	Audio .....	238
5.4.7	Powerpoint .....	239
5.4.8	Notch Blocks .....	239
5.4.9	PSD-Files .....	246
5.4.10	Live Input .....	246
5.4.11	Generative Patterns .....	254
5.4.12	HTML-Content .....	260
5.4.13	Text .....	264

5.5	<i>Create your first project</i> .....	266
5.5.1	<i>Manage Content</i> .....	269
5.5.1.1	<i>User Properties</i> .....	270
5.5.1.2	<i>Collection</i> .....	274
5.5.1.3	<i>Groups</i> .....	277
5.5.1.4	<i>Versioning</i> .....	280
5.5.1.5	<i>WatchFolder</i> .....	282
5.5.1.6	<i>Target Systems</i> .....	287
5.5.2	<i>Project Settings</i> .....	287
5.5.3	<i>Load and Save Project</i> .....	288
5.6	<i>Multi System Configuration</i> .....	292
5.6.1	<i>Session Management</i> .....	293
5.6.2	<i>Data Sync</i> .....	306
5.6.3	<i>Playback Sync</i> .....	307
5.6.4	<i>Content Sync</i> .....	311
5.6.5	<i>Data Routing</i> .....	321
5.6.6	<i>Backup</i> .....	322
5.6.7	<i>Advanced Settings</i> .....	329
5.7	<i>System Output Setup</i> .....	330
5.8	<i>Playback Live and/or in Preview</i> .....	334
5.8.1	<i>Playback Mixing Engine</i> .....	334
5.8.2	<i>Preview</i> .....	340
5.8.3	<i>Fullscreen Renderer</i> .....	342
5.8.4	<i>Output-Stream</i> .....	347
5.8.5	<i>Audio Playback</i> .....	353
5.8.6	<i>Sync Clock</i> .....	354
5.9	<i>User Interface</i> .....	358
5.9.1	<i>Top-Bar</i> .....	362
5.9.2	<i>Status Bar</i> .....	367

5.9.3	Project Explorer .....	369
5.9.4	Render Editor .....	377
5.9.5	Playback Editor .....	383
5.9.6	Inspector .....	392
5.9.7	Playbacks .....	403
5.9.8	Programmer .....	405
5.9.9	Library .....	410
5.9.10	Lock Screen .....	412
5.10	Value Mode and Programmer Mode .....	413
6.	Advanced Features .....	417
6.1	Autostart .....	419
6.2	Color Correction .....	421
6.3	Color Grading .....	423
6.4	Console Layer .....	425
6.5	Control View .....	444
6.6	Devices .....	456
6.6.1	DMX Input Device .....	464
6.6.2	DMX Output Devices .....	468
6.6.3	Powerpoint Remote .....	476
6.6.4	Stream Deck .....	483
6.6.5	Vertex NDI Streamer .....	489
6.7	DMX-512 .....	496
6.7.1	DMX Routing .....	496
6.7.2	DMX IO .....	507
6.7.3	DMX Control .....	509
6.7.4	DMX-Output .....	542
6.8	Interaction .....	551

6.8.1	Wiring Editor .....	551
6.8.2	Trigger Editor .....	556
6.9	Node System .....	559
6.9.1	Pulse(s):Polling, Push, Pull .....	563
6.9.2	Node System Editor .....	570
6.9.3	Node System Monitor .....	578
6.9.4	Working with Nodes .....	583
6.9.5	Parameters and Conditions .....	595
6.9.6	Composite Node .....	604
6.10	Presets .....	609
6.11	Schedule Editor .....	616
6.12	Scripting .....	618
6.12.1	Vertex Scripting .....	619
6.12.2	Vertex Script Examples .....	633
6.12.3	Script Editor and Script Monitor .....	645
6.12.4	Scripts .....	655
6.12.5	Other Scripting Languages .....	660
6.12.5.1	Lua Scripting .....	660
6.12.5.2	Python Scripting .....	661
6.12.5.3	C# Scripting .....	661
6.12.6	Variable .....	664
6.13	Timer .....	668
6.14	SMPTE .....	673
6.14.1	SMPTE IO Interface Configuration .....	675
6.14.2	Receiving Timecode .....	676
6.14.3	Sending Timecode .....	680
6.15	Unreal Engine Plugin .....	683
6.16	Vertex Remote Script API .....	684

---

6.17	<i>VIOSO AutoCal</i> .....	688
6.18	<i>Webserver</i> .....	691
6.18.1	<i>HTML5 Server Custom Pages</i> .....	691
7.	<i>Misc.</i>	696
7.1	<i>List of Script Commands</i> .....	698
7.2	<i>VERTEX Data Formats and File Suffixes</i> .....	711
7.3	<i>Keyboard Shortcuts</i> .....	712
8.	<i>VERTEX NDI Streamer</i>	713
9.	<i>VERTEX Transcoder</i>	721

*What is Vertex?*



# 1 What is Vertex?

## VERTEX

- *is a real-time AV Production Suite.*
- *offers a multi user work environment for visual and audio content, programming devices and management of control data.*
- *has got a discrete preview function that allows for editing and monitoring your show independently from the main output(s) - regardless of time or space*
- *is a versatile toolbox making your daily work easier and, above all, quicker.*

## VERTEX Principles

- *Focus on a quick and easy workflow for everyday standard tasks.*
- *VERTEX takes care of your project data and its sharing - even in multi client environments. So you can focus on your real work and don't have to worry about keeping your data up to date.*
- *Clean UI for basic tasks. Additional editors and settings exist when higher complexity is needed.*
- *Cuelists vs timelines - so what? Choose your preferred type and workflow.*
- *Use 2D and/or 3D previews - VERTEX suits your project's needs. Why working in a 3D environment when there is no need for that in your project?*
- *Prepared for complex task and control logic: link devices, stream media and add a logic - from easy level up to advanced installations.*
- *VERTEX accompanies you from simple to complex projects.*

## This Quick Start Guide

- *will help you to set up your first system.*
- *will give you the first rough insights about the basic UI and system principles of VERTEX.*
- *will guide you to make your first project and play out your first video.*

*Welcome*

## 2 Welcome

A big welcome to VERTEX.



### **Mobile Navigation**

*Do you read this Manual on your mobile phone?*

*Just **swipe from left to right** to open the topic tree and the navigation*

## Version and publishing state

*This Manual is based on Vertex Version: 2022 R3 and the initial Version 2020 Q1*

*Last changes to this manual: 6/12/2023*

*Copyright: © 2023 ioversal international GmbH*



*This VERTEX User Manual is still work in progress.*

*We are going to add new topics from time to time.*

*If there are is any missing information, please drop us an email with your "How-to-do-this-in-Vertex"-Question to [support@ioversal.com](mailto:support@ioversal.com)*

## Bug or Feature request

*VERTEX is carefully tested by the ioversal team and many beta users.*

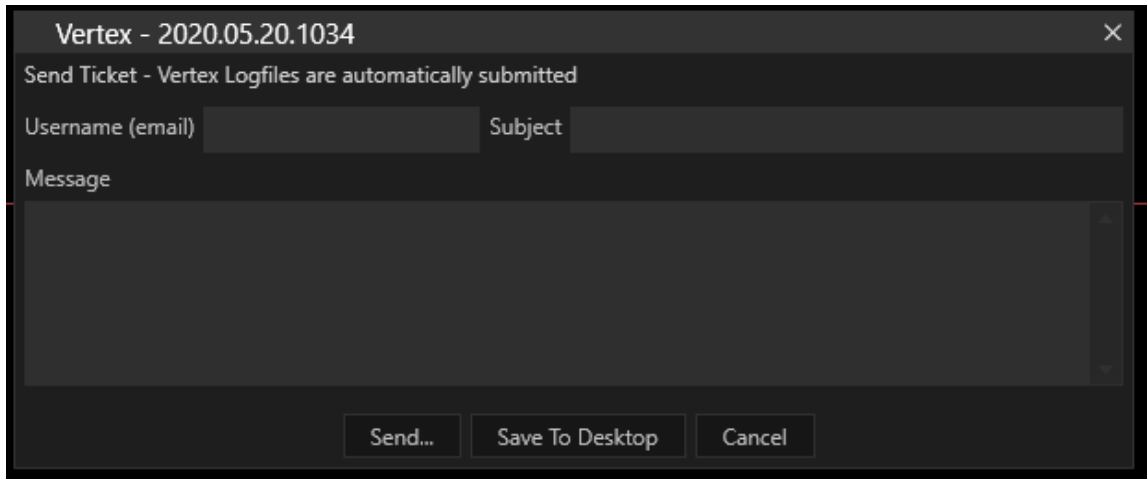
*Nevertheless bugs can occur. Or maybe you have a feature request that should be implemented into VERTEX.*

*Please report your bugs and feature request and help us to make VERTEX better.*

*Just drop us an e-mail to [support@ioversal.com](mailto:support@ioversal.com).*

*Or use the "Send Ticket" Window and create a new support ticket directly out of VERTEX.*

*This has the advantage that log files and error messages will be automatically sent as an attachment.*



*Open "Help" into the Main Menu on top and select "Send Ticket"*

## Feedback on this user manual?!

*Nothing is perfect. Got feedback on this user manual? Are there things that we don't explain clearly enough? Mistakes? Topic wishes? Not working links or images?*

*Your feedback about this would be great!*

**Please contact us and write an e-mail to [support@ioversal.com](mailto:support@ioversal.com)**

## Questions?

*Still questions?*

*Our ioversal partners and our support team are ready to help.*

*Drop an e-mail to [support@ioversal.com](mailto:support@ioversal.com)*

# *Introduction*

## 3 Introduction

### Software License, Functionality and Options

[licensing information](#)

[What's on the screen? - displaying the user interface or the rendered output in fullscreen - or both running multiple VERTEX systems in a network session](#)

## 3.1 Licenses and Editions

- A VERTEX SUITE EDITION includes all features of Edit Touch and Play offering the highest flexibility.
- Cost effective editions complement the SUITE EDITION, namely EDIT, TOUCH and PLAY. They extend to their functionality is tailored to the needs of our clients with multi-system setups.
- License purchase is available online at [www.ioversal.com](http://www.ioversal.com) or through an ioversal partner.
- A license can be either authorized on a PC system or on a dongle.
- Licenses can be [activated and de-activated](#) from within VERTEX, as well as managed online in the user account settings.

### VERTEX Editions

- To give our clients the highest flexibility in planning their projects, we offer **various Vertex editions**.
- Based on Project feedback, those three editions focus on special application scenarios and complement the VERTEX SUITE.



#### **One single Installer for all VERTEX editions**

Version handling made as easy as possible - you still need one single installer.

The feature and output restrictions of those editions lies in your license key. You have full flexibility to run a VERTEX installation as Trial, Suite, Touch, Edit or Play

Just change the license on the same system without any re-installation

### Vertex SUITE

Vertex SUITE offers the full feature set of the AV Production Suite.

Vertex SUITE is the go-to solution when flexibility is required, and offers all of the Vertex features with no limitations.

### Vertex EDIT

The perfect tool for controlling and managing your Vertex installations. EDIT includes all features except fullscreen rendering. Streaming outputs are supported.

### Vertex PLAY

A playout-only solution that can be licensed per output as a rendering engine in a larger session or as a standalone solution for high-quality, synchronized playback.

## Vertex TOUCH

The front-end tool to Vertex to display beautiful touch interfaces which can be built in Vertex EDIT or SUITE and can be used on any Windows 10 device, either standalone or in a Vertex session.

	SUITE	EDIT	TOUCH	PLAY
Fullscreen Rendering	✓	✗	✗	✓
Streaming Outputs	✓	✓	✗	✓
Fullscreen Touch UI	✓	✓	✓	✓
Project Editing	✓	✓	✗	✗
Device Control	✓	✓	✓	✓
Stand Alone	✓	✓	✓	✓
Session Mode	✓	✓	✓	✓
Vioso Calibration	✓	✗	✗	✓

## Update-Plan and Renewal

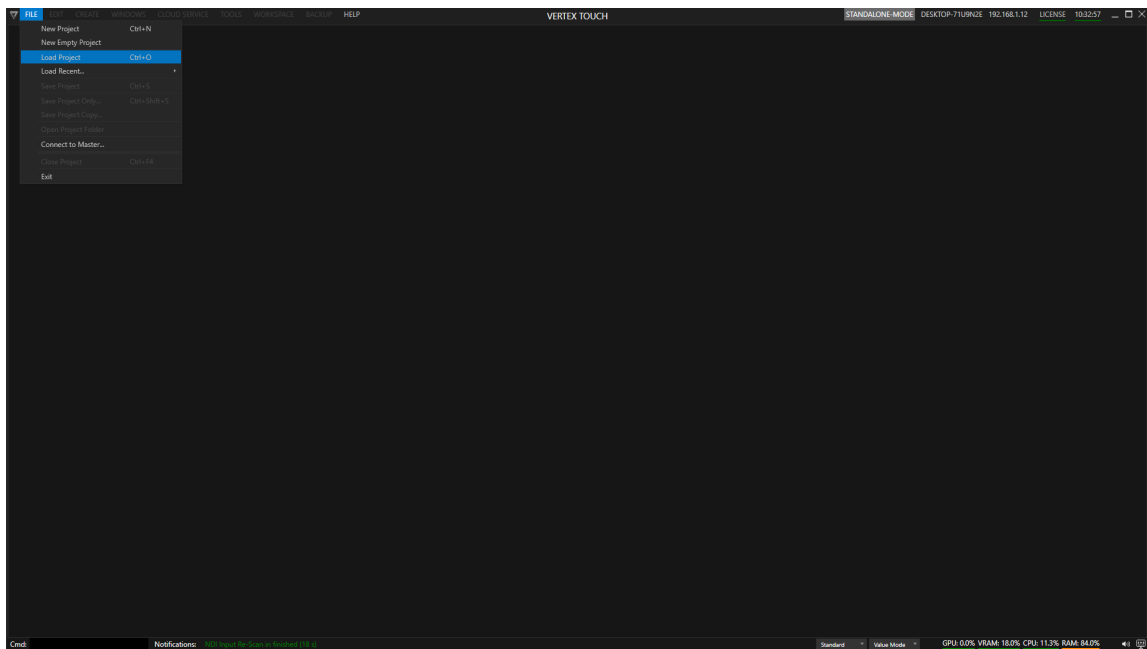
- from activation date, every license is automatically subscribed to an **update plan for 1 year**.
- the end of such a 1 year update subscriptions is called the **expiry date of a license**
- when the expiry date is reached, a license will still work for all software versions up until this point.
- however, all software updates after the expiry date will only run in TRIAL mode until the subscription to VERTEX's update plan is renewed.



- a renewal to a licence's 1-year-update-plan subscription can be purchased at any time.

## VERTEX Touch application

- VERTEX Touch is a VERTEX software version with a **reduced functionality and reduced user interface** - optimized for fullscreen touch interfaces
- The application starts **with a smaller UI footprint** without a Render Editor and no option for a Fullscreen Renderer - it is **optimized even for less powerfull PCs**
- VERTEX Touch is **shipped with every regular VERTEX Installer** - a shortcut is created as Windows application during the regular VERTEX installation process.



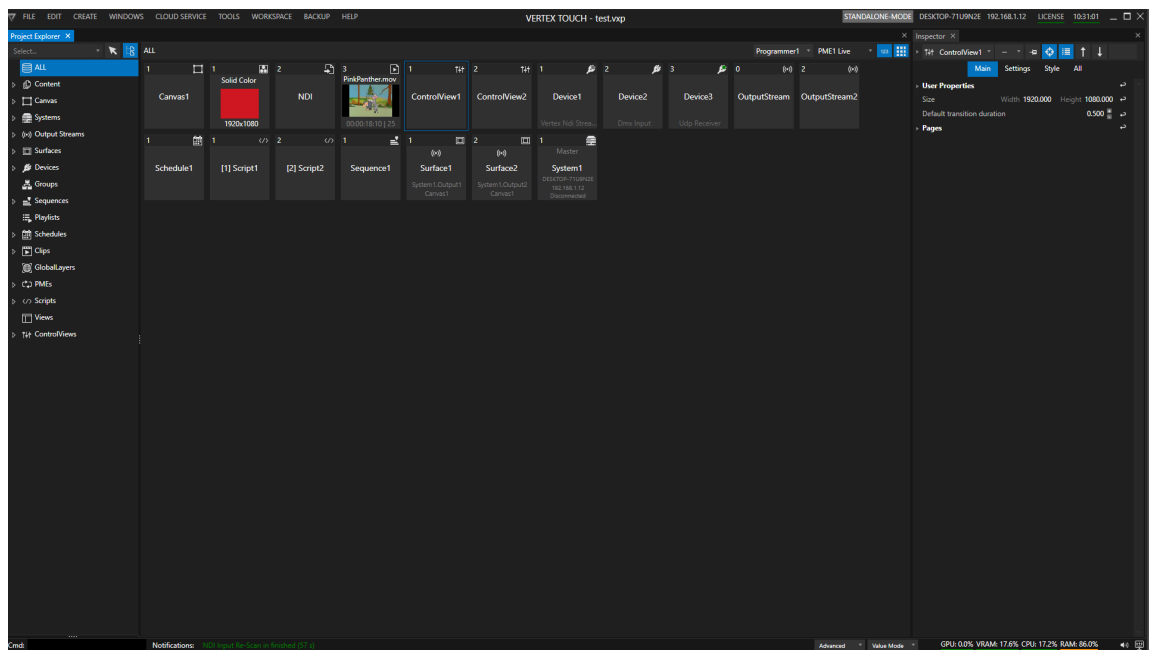
*User Interface after VERTEX Touch application was started.  
Compared to the regular VERTEX application there is no startup page*



### VERTEX Touch License Edition

ioversal also offers a [license edition](#) called VERTEX Touch for a reduces price.

You are able to run the VERTEX Touch application with a license of every other [VERTEX edition](#).



User Interface VERTEX Touch

## 3.2 User Interface and Fullscreen Renderer

There are several ways VERTEX can be displayed on a PC system:

- Per default, VERTEX shows the user interface for editing and programming when a new project is being created.
- In [fullscreen mode](#) VERTEX only displays the rendered output(s) without the UI being visible.
- Also possible and not uncommon is a [combination of the two](#) - for instance UI on screen 1 and rendered output on screen 2.

The following options are depending on your license model and system hardware:

- VERTEX SUITE offers full flexibility in setting and/ or combining UI and FS.
- EDIT, PLAY & TOUCH Editions only work in the assigned context with the sole purpose they are named after.
- Complex projects require multiple systems at times, connected in a so-called session. Go to [Multi-Systems Design](#) to learn more.

## 3.3 Multi-System Application

VERTEX is designed to work as a swarm of individual systems where every information and setting is shared.

- A **network / cluster** of various systems connected **is called a VERTEX Session**.
- There is always one system responsible for Project data handling in a so called **Master Role**.
- All other Systems are called **Session Members** that can work in UI mode or in fullscreen mode (or in both)
- **The multi-systems design** of VERTEX is **based on individual components** handling different tasks.

For an in-depth overview, please read the chapter [Multi System Configuration](#).

# *Installation and Setup*

## 4 Installation and Setup

[Hardware Requirements](#)

[Installation](#)

[Trial Version](#)

[License Activation and Deactivation](#)

## 4.1 Hardware Requirements



VERTEX is build to run on a 64 bit Windows platform only to ensure optimal performance for your System.

**Platforms that are running on 32 bit CPU architecture are not supported.**

- Windows 10 ( Recommended: Pro or Enterprise) /
- 64Bit CPU/ OS platform
- Network Card (Ethernet strictly recommended!!) Minimum 1gb - 10gb recommended for e.g. content transfer
- Graphic Card (Pro Level for complex rendering tasks, Gaming level for standard tasks)
- free choice of your hardware:
  - low for standard playout tasks (e.g. kiosk modes or digital signage Systems)
  - high performance hardware for live event or complex task
- Additional ioversal io interfaces for DMX-512 or Timecode
- SSD or even better NVMe Drive for Content handling and System folder

## Recommended Hardware Requirements

### CPU

- VERTEX needs a **balance between single-core speed and amount of CPU Cores**. Ideal are single-core speeds above 4GHz and 8 Cores and more.
- We do not recommend a dual CPU. **Avoid dual CPU Systems** - opt for a single CPU with high core speed

### GPU

- VERTEX requires a **GPU compatible with Direct X11** or higher. Nvidia Quadro Cards are recommended for demanding projects with multiple outputs per System, but Nvidia GeForce Series or AMD work as well.
- When working with a higher number of outputs or high resolution, try to test a NVidia Mosaic (or AMD Eyefinity) setup. Our internal tests indicates that these drivers perform significantly better on high number of screens or high resolutions as the OOTB-Windows solution.

## Storage and RAM

- VERTEX relies on fast data transfers. NVMe drives are recommended for compressed video playbacks and NVMe RAIDs for high performance uncompressed playback.
- Fast System RAM is essential to all tasks. Opt for highspeed RAM as possible

## Quick Tips

- **Consult with your local hardware partner** to find the best solution for your project
- When **using notebooks**, check that Microsoft Windows is using the **dedicated GPU** and not the internal one.
- **Notebooks acting as UI and Playout System** will only work smoothly if the **external and internal displays run on the dedicated GPU**.  
*Using Notebooks in such a setup is not recommended and can lead to bad performance*
- Using **multiple NVIDIA Quadro cards is possible** but **requires special skills** for setup.
- Check your Systems **energy saving settings** to ensure that **no processor throttling** is active
- VERTEX supports standard audio devices as well as ASIO Devices. **For reliable synchronization use ASIO enabled audio devices**
- Compressed Video Formats such as H264 and Apple ProRes depend more CPU performance than on the disk speed while uncompressed formats such as image sequences require fast storage

## Virtual Machines



**Licenses are not working on virtual machines and virtual desktops, except Bootcamp on Apple Hardware**

The VERTEX License System does not support software installations that run on virtual machines.

It is **not possible to activate** a VERTEX License when Windows 10 is hosted by **Virtual Box, VMware and Parallels**.

**Apple Hardware:** When Windows is started with **Bootcamp**, the License is not blocked and VERTEX should run.





## 4.2 Installation

- *How to install VERTEX on your PC.*
- *Learn about the required **Windows components** for VERTEX.*
- *Check if **VERTEX Background Services** are running after your installation and even better: **reboot your System once.***

### Install VERTEX



#### **SAME INSTALLER FOR TRIAL VERSION AND LICENSED VERSION**

Whether you want to try the VERTEX trial version or buy a license: the installer is always the same.

You can convert the trial version to a licensed version with a dongle or license activation.

If a licensed VERTEX version has been previously installed, your license will be inherited.

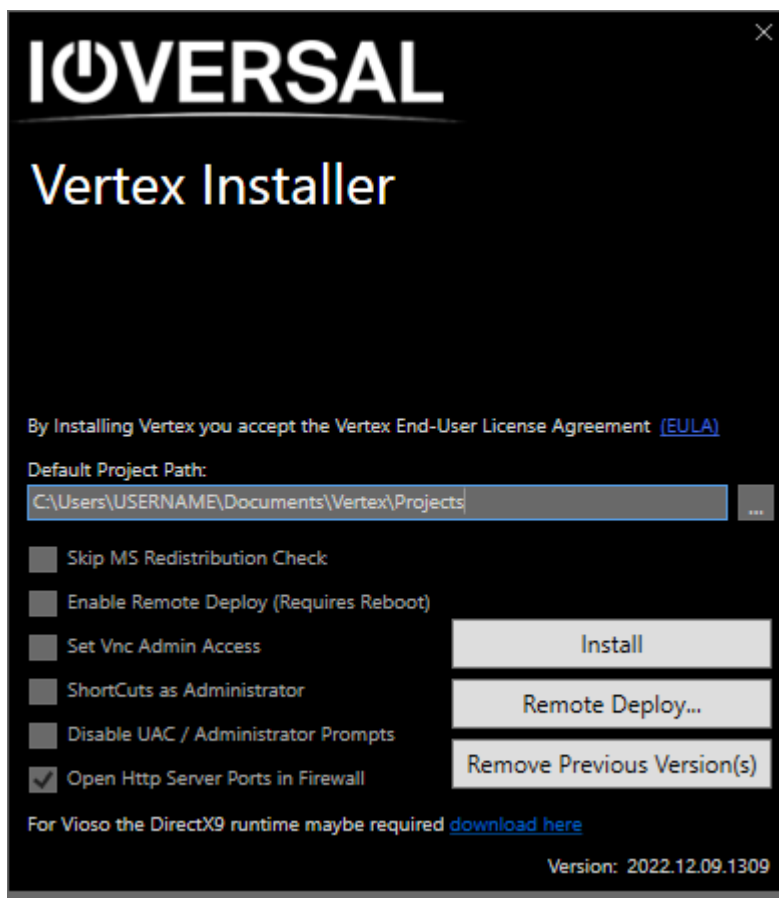


VERTEX comes with some third party components that need **C++ Redistributable** packages (2008, 2012 and 2015-2019) for which the installer will search your PC.

If none of them have been previously installed, the installer will ask you for each of the 3 packages to install.

**After each Redistributable package is installed, please continue by pressing the "Install" Button again.**

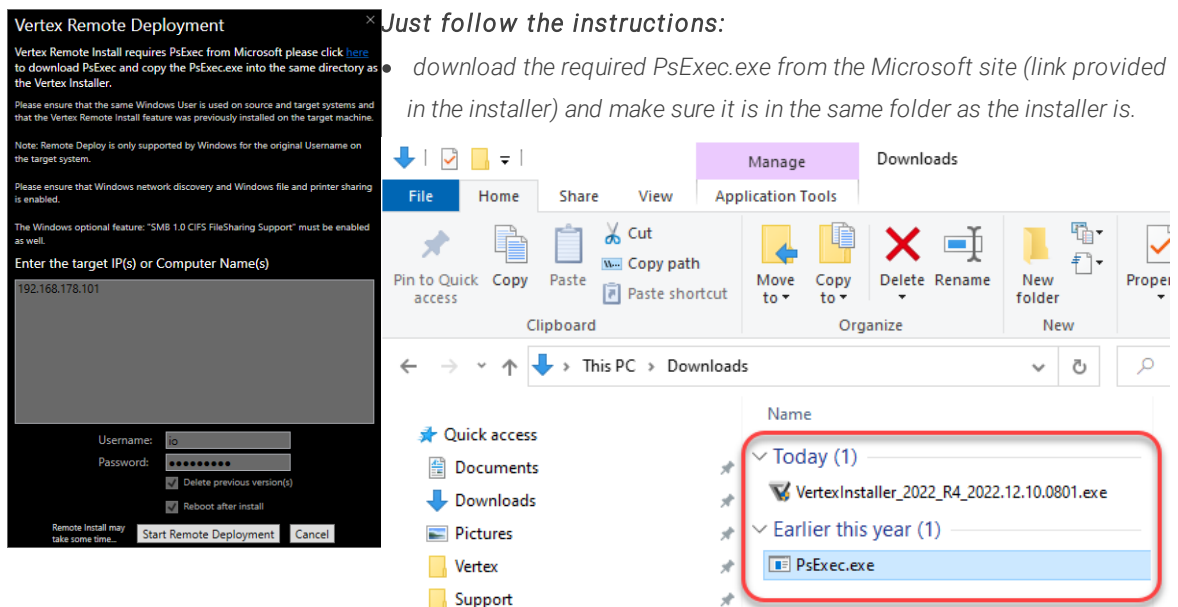
1. **Download** the latest VERTEX version from [www.ioversal.com](http://www.ioversal.com)
2. **Start the VertexInstaller.exe** on your PC
3. After the installer has been launched, please **double check** the Default Project Path.  
We strongly recommend **using a fast drive for the project path.**  
You'll be able to change this path at a later point. To avoid performance issues, always set the project path to a fast drive.
4. Follow the instructions. The installer will give you feedback on the installation progress.



Skip MS Redistribution Check	You may skip the check for C++Redistributable packages, if Vertex has been previously installed on this PC.
Enable Remote Deploy	If you need to install Vertex remotely in your network, check this box and the required components for Remote Deploy will be installed.
Set VNC Admin Access	When a VNC server is not in admin mode, dialogue windows will not be popping up, suppressing the user's access.
ShortCuts as Administrator	Enable and VERTEX's desktop shortcuts will always run in admin mode.
Disable UAC / Administrator Prompts	Disabling the User Account Control is not recommended. All running applications will gain full admin access which can be a high security risk.
Open Http Server Ports in Firewall	Required for the use of VERTEX's WebView feature.

## Remote Deploy

When using a setup with multiple systems, Remote Deploy offers a quick way to update VERTEX remotely from the master system onto its session members within the network - without having to copy or download an installer onto the other systems. To launch remote deployment, click on Remote Deploy... in the installer window. A second installer window will appear:



- Remote Deploy only works if this feature's checkbox has been enabled at a previous installation on the target system. Meaning you'll need to install VERTEX the conventional way from an installer on the target system at least once before remote deployment.
- Windows network discovery and file & printer sharing needs to be enabled, as well as the optional SMB 1.0 CIFS Support.
- Enter either the target system's name or IP and provide the installer with its username and password.



*Pro tip: There is no progressive status bar during a remote installation, so we recommend enabling Reboot after Install. Therefore you'll know when the system restarts, that the remote installation has been a success.*



**Check if .Net 4.8 is installed on your Windows Built - especially when using Windows LTSB (Long Term Service Branch)**

VERTEX requires .Net 4.8 version to start. This version is shipped with Microsoft Windows Functionality Updates.

When using Windows 10 LTSB or deactivated Updates, please double check if this .Net-Version already is installed



**VERTEX program folders**

The default installation folder for all program data is the Windows program folder: C:\Program Files\ioversal\VERTEX

The folder where all app settings are stored is C:

\Users\Public\Documents\ioversal\VERTEX

## Check Firewall and Antivirus Settings

- The VERTEX installer registers all necessary ports in your **firewall** - please double check after installation if VERTEX has access and none of the ports are blocked

**When using an Antivirus Software or/ and a firewall - take care of the following points:**

- **Check if none of the files out of the VERTEX program folder is blocked** by your Antivirus-Software.  
VERTEX is carefully designed but we unfortunately can not forecast the behaviour of each Antivirus Software on the market
- **Check the *firewall settings* and allow VERTEX and its services to communicate both in private and public networks**

## Check Windows Update and Notification Settings

**Windows 10 Pro or Enterprise for Production Environments!**

Although VERTEX runs both on Windows 10 Home and Pro, we strictly recommend to use Windows 10 Pro for production environments.

Windows 10 Pro comes up with more detailed settings especially for maintenance updates. Windows 10 Home is shipped with an automated update service. There is no reliable and permanent option to be in control of those updates in the Home Version, whereas the Pro version is customizable in regard to updating preferences.

## Check if VERTEX Background Services are running

- **VERTEX Background Services** are a **group of Windows Services** that are required to run in the background ensuring connections between different VERTEX system.
- The services start running as soon as the Windows user is logged in.

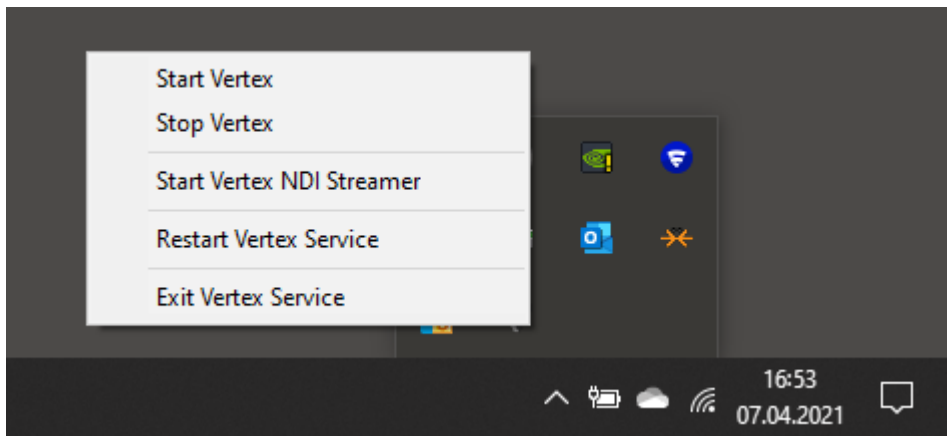


Please do not change any settings on Windows 10 services -

The VERTEX background service must run on Windows startup to ensure communication in a multi client setup.

## Start and Stop Background Service

- You can start/stop and restart all VERTEX Background Services with the menu that opens after a right-click on the VERTEX tray icon in your Windows task bar.



- *If you run the installer to update a licensed version of vertex, the newer version will run under your license unless your license's update plan has expired. Then VERTEX will automatically run in trial mode. In this case, you may go back to your older version or you may contact us to renew your update plan. Also, it is possible to have multiple versions of VERTEX installed.*

## 4.3 Trial Version

- **Download** a VERTEX Trial Version on [www.ioversal.com](http://www.ioversal.com)
- The Trial version has the **full functionality** with all features and **only some little playback and network limitations**.
- Every VERTEX Trial Version **can be converted to a full version** of the software. Just [log in with a valid VERTEX license](#) or plug in an activated dongle.

## Features

- A **project file** that has been saved with a VERTEX Trial version **can be opened with a licensed version**
- **Session Test Mode:** Trial Versions are able to connect with Trial Versions to test a Multi-Client Session
- **No limitation in functionality and features**, no limitation for number of playbacks or bandwidth

## Limitations

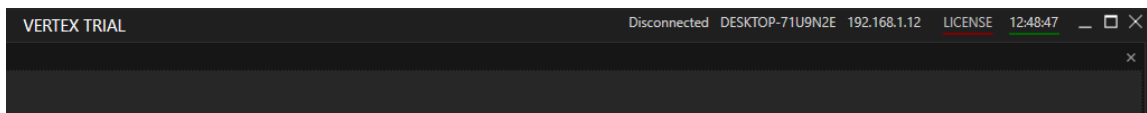
- "VERTEX Trial" is displayed in the **Top Bar**
- In fullscreen mode and in Render Editor **a watermark** is displayed every 10 Minutes
- Playbacks stop every 10 Minutes
- The **outputs** for external protocols ( e.g Art-Net™ ) stop every 10 Minutes
- Trial Versions are **only allowed to connect with Trial Versions** to test a **Multi-Client Session**. A Session with licensed VERTEX Systems is not possible. The network connections to licensed VERTEX versions are cut after a few seconds.



## 4.4 License Activation and Deactivation

- There is a **Menu for License Management** which is located in the **top right corner of the [top bar](#)**
- Check the **Status, Activate or Deactivate** your **System License** or a **Smart Dongle License**
- Read the topic [License model and Options](#) to get all information about the **different types of VERTEX licenses**

### License Menu



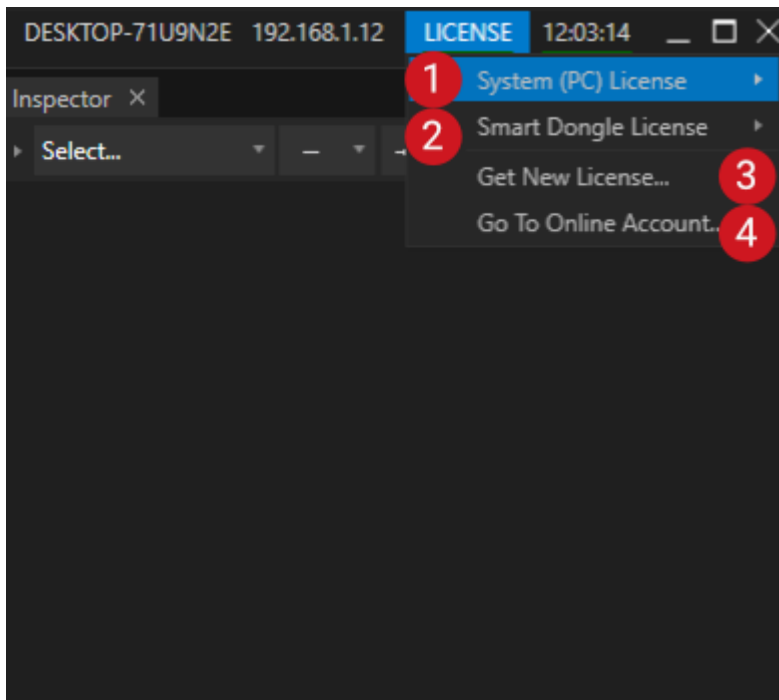
A **green or red line** under **LICENSE** quickly shows you the **license status**:

- if a **valid license is activated (green)** or
- if **no license is activated (red)** and VERTEX is in [Trial Mode](#)

The **license menu** navigates you through the **different options** to handle, activate and deactivate your license.

You can easily do this with an **internet connection**, but there are also **offline options** to do this without internet.

Both for a **System (PC) License** and a **Smart Dongle License**

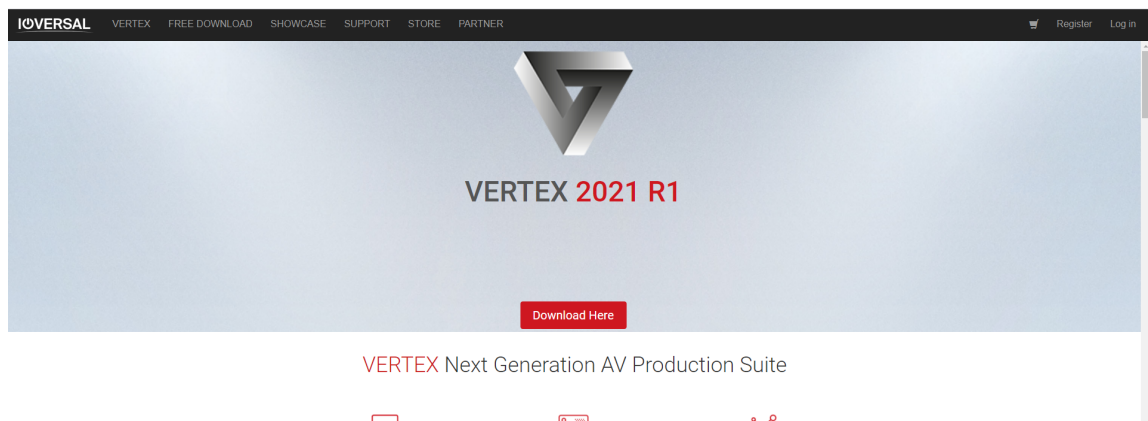


<p><b>1</b></p>	<p><b><u>System (PC) License</u></b></p>	<p><b>Activate</b> and lock one of your VERTEX license on your PC  <b>Deactivate</b> and unlock licenses from your PC  <b>Update</b> Licenses</p> <p>All options are possible:  <b>online</b> - connected to the internet (fast and easy)  <b>offline</b> - with a second device with an internet connection (takes a little longer, but the mediaserver PC can stay offline and without internet)</p>
<p><b>2</b></p>	<p><b><u>Smart Dongle License</u></b></p>	<p><b>Activate</b> and lock one of your VERTEX license on one of ioversals Smart Dongles  <b>Deactivate</b> and unlock licenses to your Smart Dongle  <b>Update</b> Licenses</p> <p>All options are possible:  <b>online</b> - connected to the internet (fast and easy)  <b>offline</b> - with a second device with an internet connection (takes a little longer, but the mediaserver PC can stay offline and without internet)</p> <p>You can buy <b>Smart Dongles</b> in the <a href="#">ioversal online store</a> or at one of <a href="#">our partners</a></p>

<p><b>3</b></p>	<p><b>Get new License</b></p>	<p>Jump to the ioversal online store. Instantly buy a new license. If paid with e.g. Credit Card the new license will be able to use within seconds.</p>
<p><b>5</b></p>	<p><b><u>Go to Online Account</u></b></p>	<p>Go to the ioversal online account and login there to see all licenses that are connected to your account</p>

## Online Account

- You are able to instantly buy licenses online
- all your licenses are managed into your **ioversal online account**
- just **log in on [www.ioversal.com](http://www.ioversal.com)**



- Once logged in, **click on your user name** to see different options
- **Click on "Licenses"** to see and manage all your VERTEX licenses

## My Account



Licenses



Serials

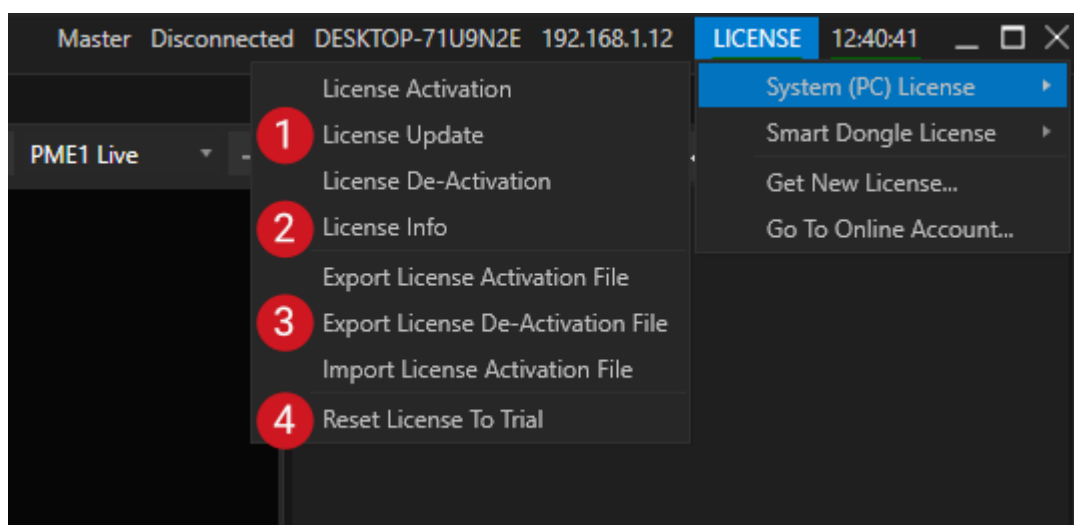


Account Settings



My Support

## System (PC) License



1	<a href="#">Online</a>	<b>Activate</b> VERTEX and lock your license to the Hardware
---	------------------------	--

	<a href="#"><u>Activation, Deactivation or Update</u></a>	<p><b>Deactivate</b> VERTEX: Unlock your license from a Hardware and use it on another</p> <p><b>Update:</b> Update your current license. e.g. when a renew for another feature update period</p> <p>For all this options, the PC with VERTEX has <b>to be connected to the Internet.</b></p>
2	<a href="#"><u>License Info</u></a>	Shows all information about your current license ( <b>user, expiry date, license key, type..</b> )
3	<a href="#"><u>Offline Activation/Deactivation</u></a>	<p><b>Activate</b> VERTEX and lock your license to the Hardware</p> <p><b>Deactivate</b> VERTEX: Unlock your license from a Hardware and use it on another</p> <p>For this options, <b>NO internet connection</b> is required on the PC on that VERTEX is running.</p> <p>But you need to have <b>a second device with internet</b> to log in to your ioversal user account.</p>
4	<a href="#"><u>Reset License to Trial</u></a>	Delete all license information from your current hardware and <b>set your license back to Trial.</b>

## License Activation/Update/ Deactivation

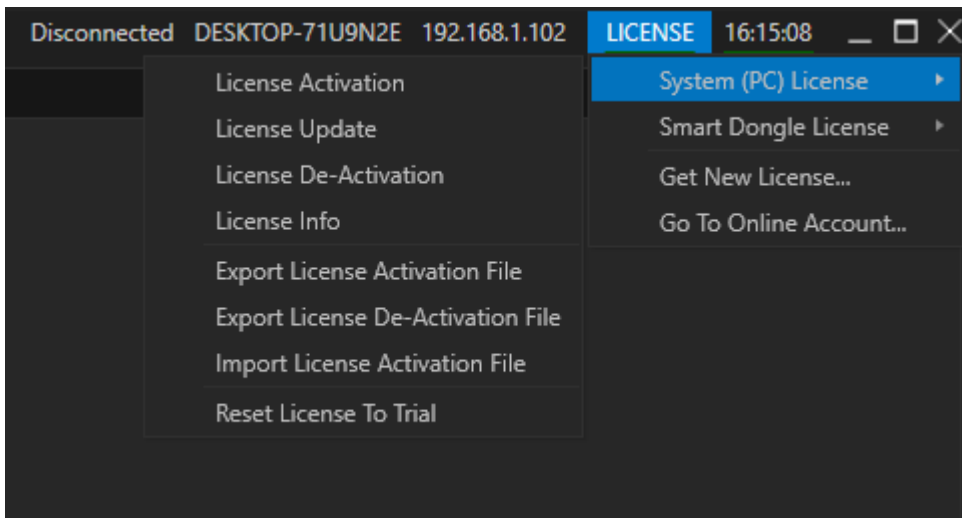


**You need to have:**

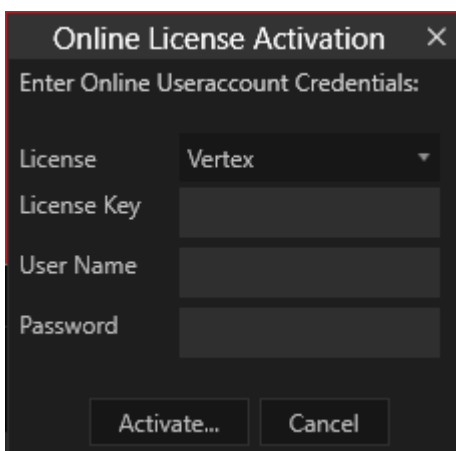
1. an user account on [www.ioversal.com](http://www.ioversal.com)
2. a valid VERTEX license with status "released" (that was bought with/transferred to this account and that is not already activated on another system/dongle)  
for a license update: a valid update that was bought and payed in your online license management
3. an internet connection

## Activation

1. **Start** VERTEX
2. Go to the license menu



3. Select System (PC) License
4. Click to "License Activation"



5. Enter your user account credentials and your password.  
*As an option:* Enter a certain **License Key**, e.g. if you manage a bunch of VERTEX licenses with your user account. With this option you are able to activate a specific license  
*If no License Key is entered*, the license manager will automatically pick up one of the free licenses from your user account
6. A status message will show you whether the activation was successful or not.
7. Wait a few seconds - The red line under License should change to green



### **Internet-Connection and License**

Once a license is successfully activated, feel free to cut off the internet access for your System again. VERTEX and the activated license will work until the Expiry Date is reached (the check the Expiry Date got to License Info in the License Menu).

When you have reached the expiry date of such an "offline" System, you are not able to install new VERTEX releases - but the old software versions are still working

## Deactivation

1. Click to License De-Activation
2. Enter your user name and your password, confirm
3. A status message will show you whether the activation was successful or not.
4. The VERTEX License Status is underlined in red
5. Your License in your online license management at [www.ioversal.com](http://www.ioversal.com) is in status released again

## Update



### **Buy an update and renew your License**

You are able to renew your license to get among others the newest updates for VERTEX versions. To Update a license, log in to your user account, go to "Licenses" there. Open your License list, click to "renew". The license update is put in the shopping cart.

If you want to change the type of your license, please [contact the support](#)

1. Check at your user account for a valid update for the license that is logged to the System you want to update
2. Go to entry "**License Update**"
3. Enter your user name and your password, confirm
4. A status message will show you whether the activation was successful or not.
5. Check at **License Info**: the displayed expiry date should be extended

## License Info

- Opens a Window that shows you all information about your current VERTEX License

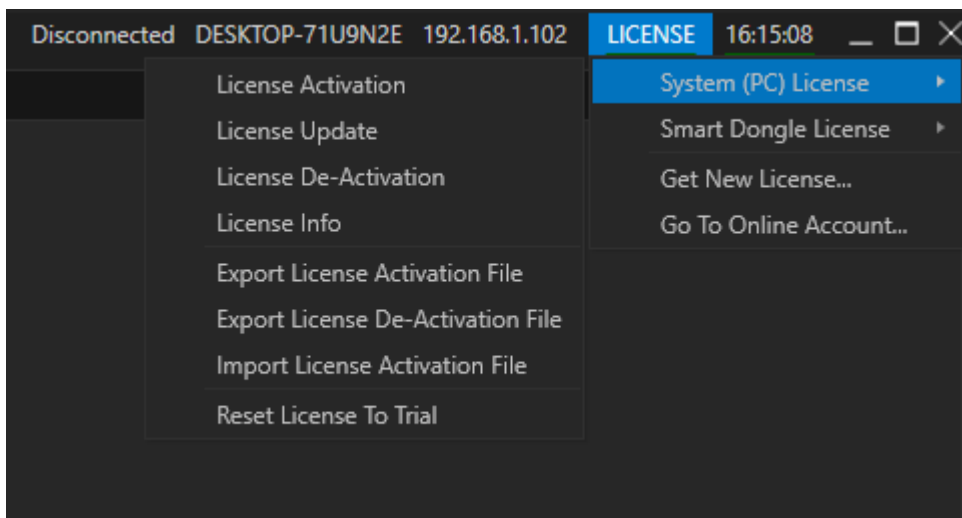
## Offline Activation



### You need to have:

1. an user account on [www.ioversal.com](http://www.ioversal.com)
2. a valid VERTEX license (that was bought with/transferred to this account and that is not already activated on another system/dongle)
3. an USB Stick, SD Card or external harddrive
4. a second PC with an internet connection

1. Start VERTEX
2. Open License Menu
3. Select System (PC) License



4. Click to **"Export License Activation File"**
5. Enter your user account credentials and your password and confirm (This information will be stored into the activation file)
6. **Save the activation file** on a portable drive
7. Go to a PC with internet connection and **sign in with your user account on [www.ioversal.com](http://www.ioversal.com)**
8. Go to "My Licenses", select "License details"



---

## License Details

Product	Vertex - License
License Key	edef897b-6e40-48c2-9b67-8edfba2abcf0
State	Inactive
Mode	Software
Expiry Date	7/15/2022
Label	<input type="text"/>
Notes	<input type="text"/>
	<input type="button" value="Save"/>

### Actions

*License details at [www.ioversal.com](http://www.ioversal.com) for a released but not yet activated license. Once activated, you will see more options for e.g. deactivation*

9. Click to **"Activate License/Dongle"**
10. **Upload the Activation file**
11. **Download the "Receipt" Activation File** and upload this to your portable Drive
12. Go back to your PC again, click to **"Import License Activation File"**
13. Choose the "receipt" activation file you have downloaded before
14. Your license should be activated now.

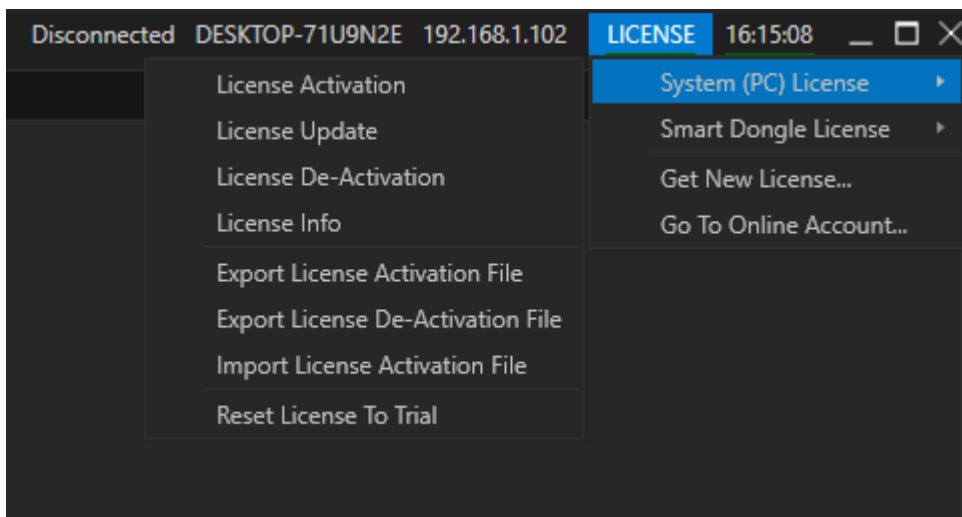
*After a few seconds the license status should change from red to green .*

## Offline Deactivation

**You need to have:**

1. an user account on [www.ioversal.com](http://www.ioversal.com)
2. A VERTEX system with an activated license which is assigned to your user account
3. an USB Stick, SD Card or external harddrive
4. a second PC with an internet connection

1. Start VERTEX
2. Open License Menu
3. Select System (PC) License
4. Select Export License De-Activation File



5. Enter your User name and your password
6. Save the deactivation file on a portable drive
7. Go to a PC with internet connection and **sign in with your user account on [www.ioversal.com](http://www.ioversal.com)**
8. Go to **"My Licenses"**, select **"License details"**
9. Click to **"Deactivate License/Dongle"**

## License Details

Product	Vertex - License
License Key	b884eb15-b9ad-4149-8516- fbb311b6735e
State	Activated
Mode	Software
Expiry Date	7/15/2022
Computer Name	DESKTOP-71U9N2E
Label	<input type="text"/>
Notes	<input type="text"/>
	<input type="button" value="Save"/>
Actions	<input type="button" value="Deactivate License/Dongle"/>
	<input type="button" value="Download Activation File"/>
	<input type="button" value="Emergency License Reset"/>

*License details at [www.ioversal.com](http://www.ioversal.com) for a license that was activated on a system before. Once activated, you will see more options for e.g. deactivation*

10. A file **upload dialog** opens
11. Select the **deactivation file and upload it**

## License Deactivation

License Key: b884eb15-b9ad-4149-8516-fbb311b6735e

Computer Name: DESKTOP-71U9N2E

To deactivate your license you can either deactivate the license from your application directly or upload a license deactivation file.

Upload Deactivation File

Select File

Upload

[Back to Details](#)

12. Your license now should be **successfully deactivated**



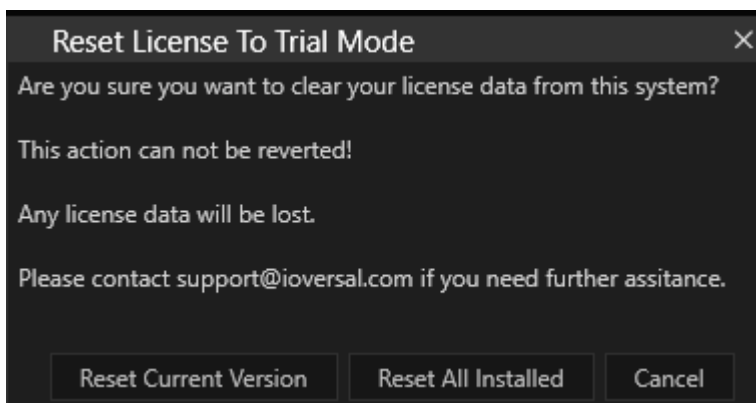
### **De-Activation file when VERTEX is uninstalled**


*If you uninstall VERTEX from your PC and the software is still activated, the VERTEX uninstaller offers you to save a deactivation file.*

*To deactivate your license, please also do steps 7 to 12.*

## Reset License to Trial

- With this option you are able to reset only the current or all VERTEX Version on your PC back to Trial mode!



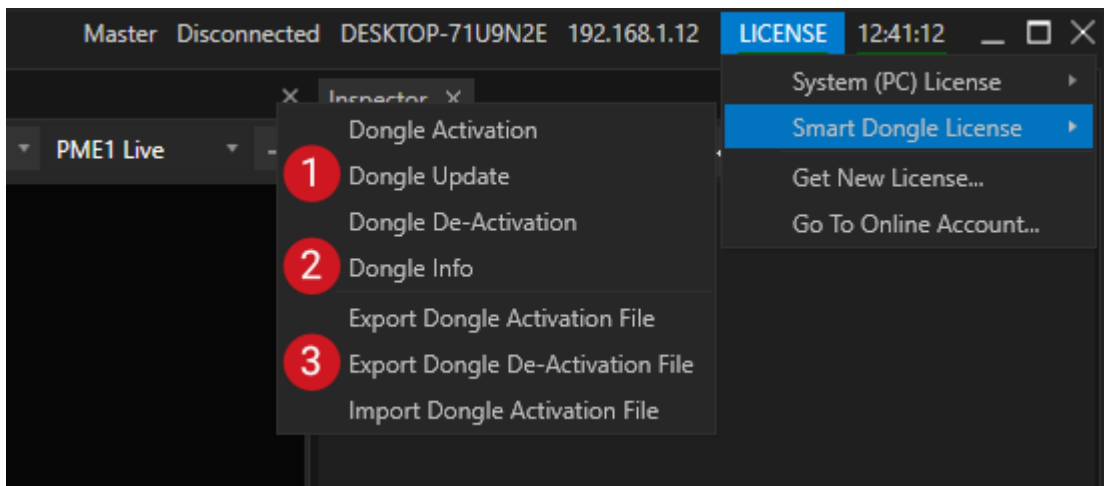


**Reset License to Trial - Licenses will not be deactivated**

Please consider that all license data is lost. If no deactivation is done before, the license still is logged on your hardware and can not used on another system.

If accidentally done, contact the [ioversal support](#) or do an [Emergency License Reset](#)

## Smart Dongle License



<b>1</b>	<p><a href="#">Dongle Online Activation/Deactivation or Update</a></p>	<p><b>Dongle Activation:</b> store and activate your license to an ioversal Smart Dongle</p> <p><b>Dongle Deactivation</b> Unlock your license from an ioversal Smart Dongle</p> <p><b>Dongle Update:</b> Update your current license. e.g. when a renew for another feature update period</p> <p>For all this options, the PC with VERTEX has <b>to be connected to the Internet.</b></p>
<b>2</b>	<p><a href="#">Dongle Info</a></p>	<p>Shows all information that is stored on your smart dongle (<b>user, expiry date, license key, type..</b>)</p>
<b>3</b>	<p><a href="#">Dongle Offline Activation/Deactivation</a></p>	<p><b>Dongle Activation:</b> store and activate your license to an ioversal Smart Dongle</p> <p><b>Dongle Deactivation</b> Unlock your license from an ioversal Smart Dongle</p> <p>For this options, <b>NO internet connection</b> is required on the PC on that VERTEX is running.</p>

		But you need to have <b>a second device with internet</b> to log in to your ioversal user account.
--	--	--

## Dongle Activation/Update/ Deactivation



### You need to have:

1. an user account on [www.iversal.com](http://www.iversal.com)
2. a valid VERTEX license (that was bought with/transferred to this account and that is not already activated on another system/dongle)
3. an internet connection
4. an empty ioversal Smart Dongle that was bought from a distributor or on the ioversal online store

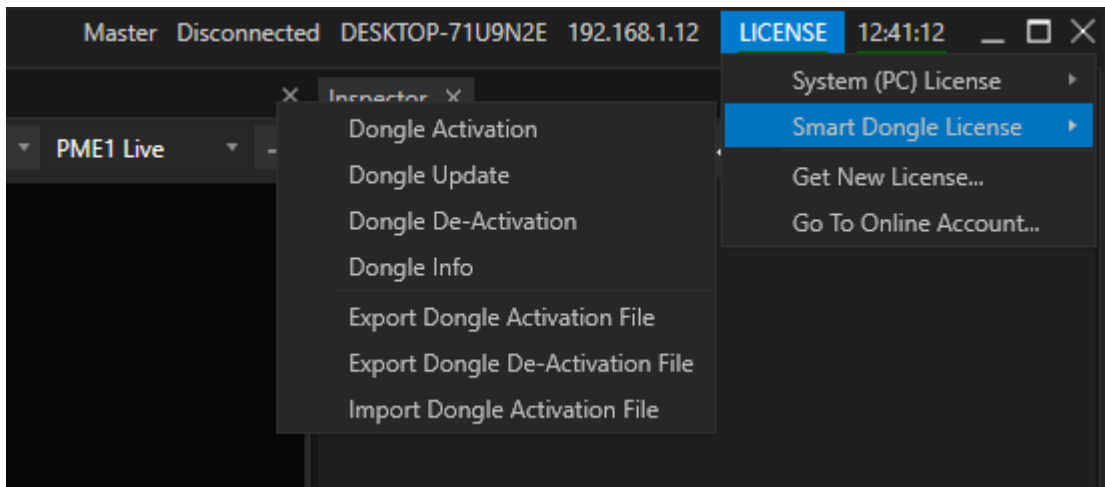


### USB 3-Hubs and Dongles

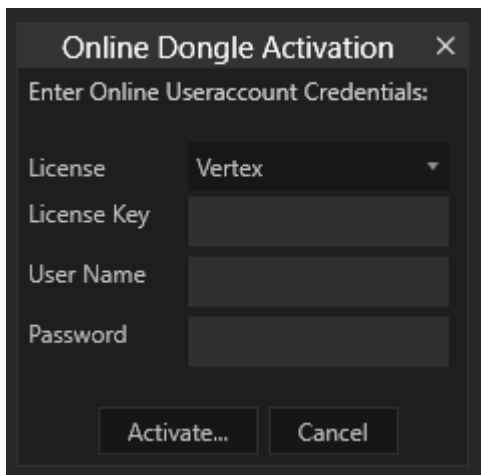
Try to avoid to plug in Dongles into an USB 3.0 Hub. Some hubs might have down compatibility problems with USB 2 and 1 drivers. Result can be that the dongle is frequently check out and in again for a short time frame. You can avoid this risk by using the USB 3.0 or USB 2.0 ports on your System to plug the dongle in.

## Activation

1. Plug in an empty ioversal Smart Dongle to one of the USB Ports of your PC
2. **Start** VERTEX
3. Go to the **license menu**



4. Select **Smart Dongle License**
5. Click to **"Dongle Activation"**



6. Enter your user account credentials and your password.  
*As an option:* Enter a certain **License Key**, e.g. if you manage a bunch of VERTEX licenses with your user account. With this option you are able to activate a specific license  
*If no License Key is entered*, the license manager will automatically pick up one of the free licenses from your user account
7. A status message will show you whether the activation was successful or not.
8. Wait a few seconds - The red line under License should change to green



### **Internet-Connection and License**

Once a license is successfully activated, feel free to cut off the internet access for your System again. VERTEX and the activated license on your dongle will work until the Expiry Date is reached (the check the Expiry Date got to License Info in the License Menu).  
Feel free to carry your activated dongle with you and to use it and your license on every

VERTEX system

## Deactivation

1. Click to **Dongle De-Activation**
6. Enter your user name and your password, confirm
7. A status message will show you whether the activation was successful or not.
8. The VERTEX License Status is underlined in red
9. Your License in your online license management at [www.ioversal.com](http://www.ioversal.com) is in status released again

## Update



### **Buy an update and renew your License**

You are able to renew your license to get among others the newest updates for VERTEX versions. To update a license, log in to your user account, go to "Licenses" there. Open your License list, click to "renew". The license update is put in the shopping cart.

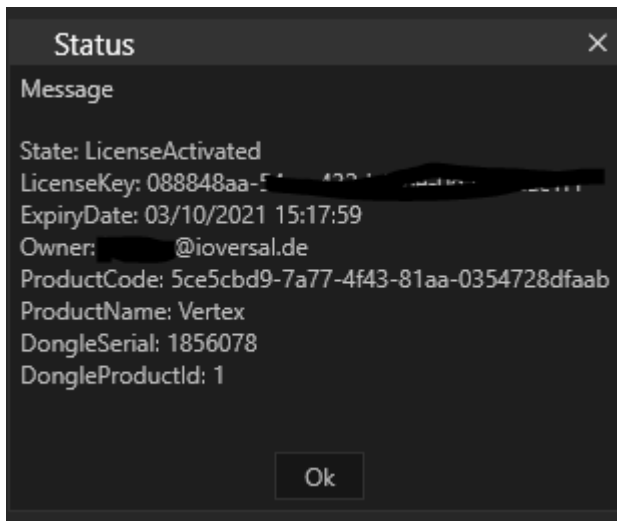
If you want to change the type of your license, please [contact the support](#)

1. Check at your user account for a valid update for the license that is logged to the System you want to update
6. Go to entry "**Dongle Update**"
7. Enter your user name and your password, confirm
8. A status message will show you whether the activation was successful or not.
9. Check at **Dongle Info**: the displayed expiry date should be extended

## Dongle Info

- Opens a Window that shows you all information that is stored on the ioversal Smart Dongle that is currently plugged in





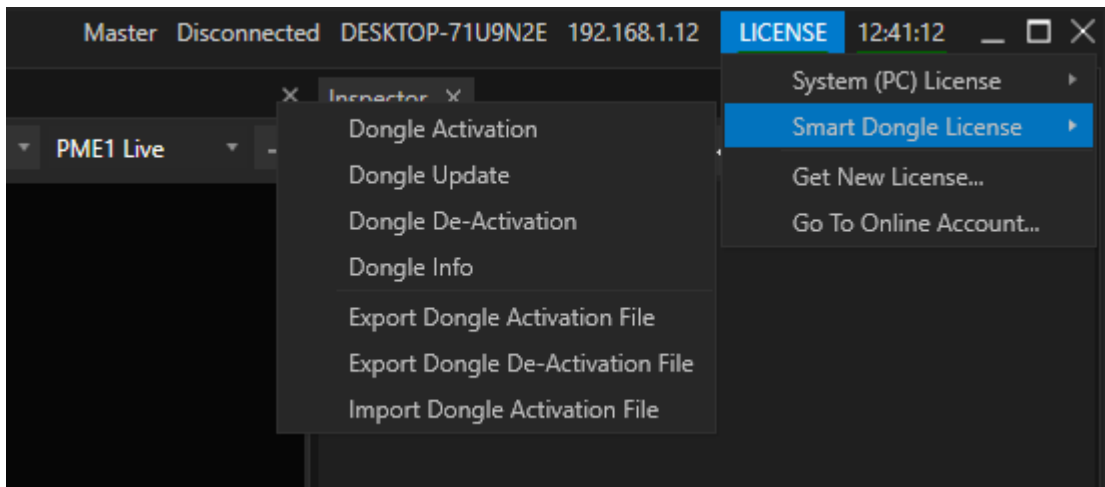
## Dongle Offline Activation



You need to have:

1. an user account on [www.ioversal.com](http://www.ioversal.com)
2. a valid VERTEX license (that was bought with/transferred to this account and that is not already activated on another system/dongle)
3. an USB Stick, SD Card or external harddrive
4. a second PC with an internet connection
5. an empty ioversal Smart Dongle that was bought from a distributor or on the ioversal online store

1. Start VERTEX
2. Open License Menu
3. Select Smart Dongle License



4. Click to **"Export Dongle Activation File"**
5. Enter your user account credentials and your password and confirm (This information will be stored into the activation file)
6. **Save the activation file** on a portable drive
7. Go to a PC with internet connection and **sign in with your user account** on [www.ioversal.com](http://www.ioversal.com)
8. Go to "My Licenses", select "License details"

---

## License Details

Product	Vertex - License
License Key	edef897b-6e40-48c2-9b67-8edfba2abcf0
State	Inactive
Mode	Software
Expiry Date	7/15/2022
Label	<input type="text"/>
Notes	<input type="text"/>
	<input type="button" value="Save"/>

### Actions

*License details at [www.ioversal.com](http://www.ioversal.com) for a released but not yet activated license. Once activated, you will see more options for e.g. deactivation*

9. Click to "**Activate License/Dongle**"
10. **Upload the Activation file**
11. **Download the "Receipt" Activation File** and upload this to your portable drive
12. Go back to your PC again, click to "**Import License Activation File**"
13. Choose the "receipt" activation file you have downloaded before
14. Your license should be **activated now on your Dongle**  
After a few seconds the license status should change from red to green.  
To double check, please open [Dongle Info](#)

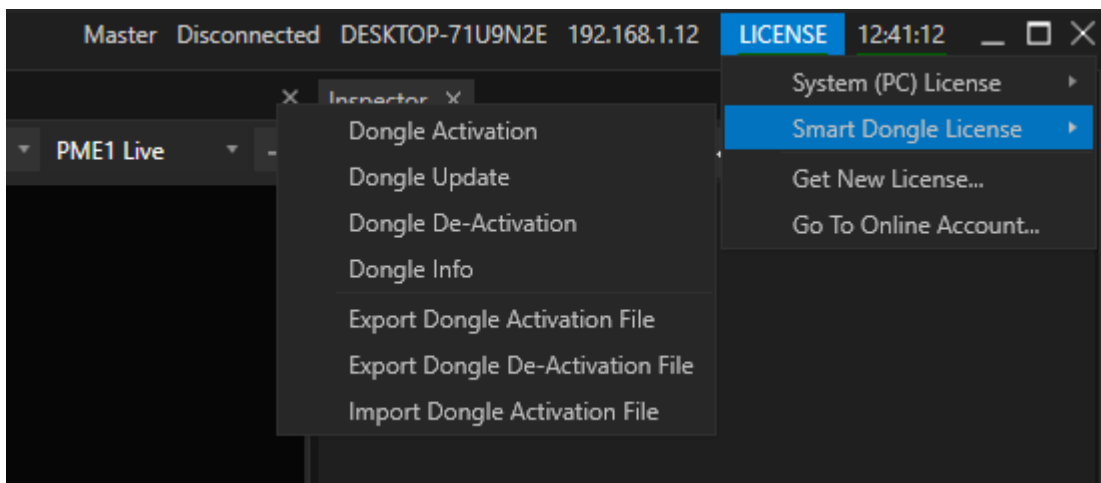
## Offline Deactivation



**You need to have:**

1. an user account on [www.ioversal.com](http://www.ioversal.com)
2. A VERTEX system with an activated license which is assigned to your user account
3. an USB Stick, SD Card or external harddrive
4. a second PC with an internet connection
5. An ioversal Smart Dongle with an activated License which is plugged in

5. Start VERTEX
6. Open **License Menu**
7. Select **Smart Dongle License**
8. Select **Export Dongle De-Activation File**



10. Enter your user name and your password
11. Save the deactivation file on a portable drive
12. Go to a PC with internet connection and **sign in with your user account on [www.ioversal.com](http://www.ioversal.com)**
13. Go to "**My Licenses**", select "**License details**"
14. Click to "**Deactivate License/Dongle**"

## License Details

Product	Vertex - License
License Key	b884eb15-b9ad-4149-8516- fbb311b6735e
State	Activated
Mode	Software
Expiry Date	7/15/2022
Computer Name	DESKTOP-71U9N2E
Label	<input type="text"/>
Notes	<input type="text"/>
	<input type="button" value="Save"/>
Actions	<input type="button" value="Deactivate License/Dongle"/>
	<input type="button" value="Download Activation File"/>
	<input type="button" value="Emergency License Reset"/>

*License details at [www.ioversal.com](http://www.ioversal.com) for a license that was activated on a system before. Once activated, you will see more options for e.g. deactivation*

13. A file **upload dialog** opens
14. Select the **deactivation file and upload it**

## License Deactivation

License Key: b884eb15-b9ad-4149-8516-fbb311b6735e

Computer Name: DESKTOP-71U9N2E

To deactivate your license you can either deactivate the license from your application directly or upload a license deactivation file.

Upload Deactivation File

Select File

Upload

[Back to Details](#)

15. Your license now should be **successfully deactivated**



### **De-Activation file when VERTEX is uninstalled**

If you uninstall VERTEX from your PC and the software is still activated, the VERTEX uninstaller offers you to save a deactivation file.

To deactivate your license, please also do steps 7 to 12.

## Emergency License Reset

- The Emergency License Reset should help you when your **hardware was changed without a license logout** before, your hardware was **stolen** or even if **something went absolutely wrong** during your license activation
- You can trigger an Emergency License Reset at your [www.ioversal.com](http://www.ioversal.com) online Account ( My Licenses -> Details)



**For every of your licenses you can trigger an Emergency License Reset maximum 3 times.**

*If you have reached this limit for somehow, please contact the support team.*

*With every Emergency License Reset the recent hardware and/or dongle ID that was connected with this license will be blocked and blacklisted. You are able to activate this license again on another System.*

## Steps

- Go to [www.ioversal.com](http://www.ioversal.com)
- Log In with your account
- Go to "My Licenses"
- select the affected License there and click on "Details"
- If this licenses is activated you will find there the Emergency License Reset Button
- Check the details and confirm

## License Details

Product	Vertex - License
License Key	b884eb15-b9ad-4149-8516-fbb311b6735e
State	Activated
Mode	Software
Expiry Date	7/15/2022
Computer Name	DESKTOP-71U9N2E
Label	<input type="text"/>
Notes	<input type="text"/>
	<input type="button" value="Save"/>
Actions	<input type="button" value="Deactivate License/Dongle"/>
	<input type="button" value="Download Activation File"/>
	<input type="button" value="Emergency License Reset"/>

*License details at [www.ioversal.com](http://www.ioversal.com) for a license that was activated on a system before. Once activated, you will see more options for e.g. deactivation*

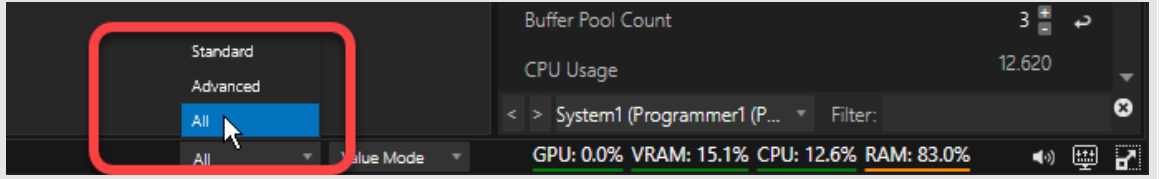


## 4.5 FAQ Performance Recommendations

As Vertex is designed to be hardware independent the software comes with reasonable variety of settings to tweak its behavior and thus matching your specific hardware's performance to the needs of your individual projects.

The initial values of the settings have been chosen to establish a reliable playback on an average scaled hardware whilst playing back average sized content in an average amount. Due to the nature of averages, you might feel the need of improving the behavior of Vertex to either slowing it down to handle a higher quantity of media on a slower system or to unleash your high performance hardware.

**Note:** Switch the inspector mode to "All" in the drop-down selection of the status bar in order to access the parameters listed below.



The screenshot shows a software interface with a dropdown menu for inspector mode. The menu options are Standard, Advanced, All, and All. The 'All' option is highlighted in blue and circled in red. The status bar at the bottom displays system performance metrics: GPU: 0.0%, VRAM: 15.1%, CPU: 12.6%, RAM: 83.0%.

### PERFORMANCE OPTIMIZATIONS:

#### Single GPU

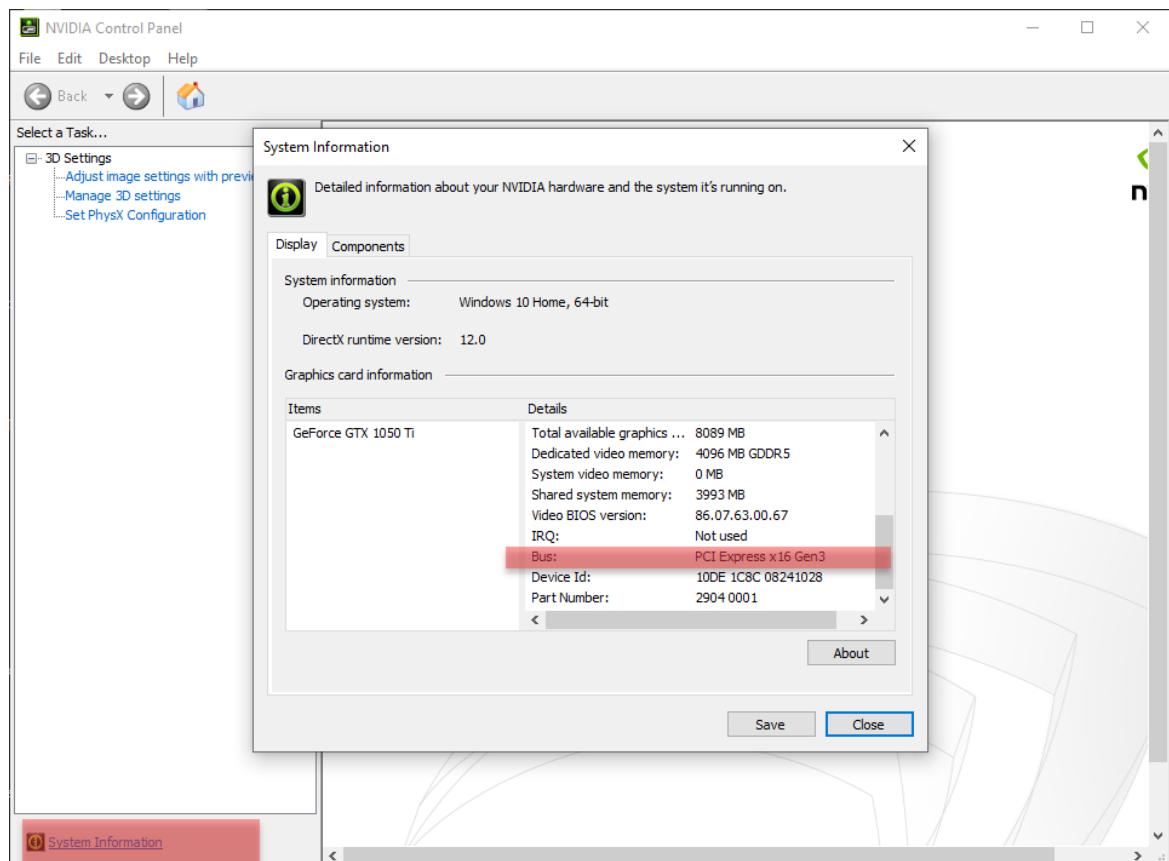
- Make sure you have the **same monitor sync frequencies and refresh rates on each output** of a system.
- Try using the **same resolution on each output** of a system.
- Ideally use one system as a master and another system exclusively dedicated to playback in fullscreen mode without the UI. Same applies for audio playback.

#### Multi GPU

- When mixing UI and fullscreen outputs try using **two GPUs - one for the UI and one for fullscreen** rendering.
- When **using multiple GPUs** ensure that the UI and fullscreen GPU are setup correctly -> **System** context menu
- Check mosaic setup to ensure the same resolutions and frequencies (EDIDs) for your multi GPU usage.

### General

- Try change the clip containers' settings to **Render In View Only**. This will ensure that content will not be rendered for those surfaces and respective outputs where it is not placed and visible. By rendering only the content in view of a surface you can free up valuable processing resources.
- Freeze the property updates of a surface or static non animated clip containers. This will reduce the amount of property updates per rendered frame. In doing so the properties are only updated when the playback enters a clip.
- Avoid too many surfaces - up to 8 work fine. If you need more, then freezing the properties might be required to reduce the rendering workload.
- Render a surface in an output directly by switching the surface render mode to output in the inspector settings of the surface - careful though, as this may lead to undesired side effects when rendering multiple surfaces into the same output as overlapping content may be the result.
- Render content directly to an output by setting the clip target of a clip container to the designated output. By doing so, you can bypass canvas-surface workflow, if necessary.
- When using GPUs on a PCIe bus, make sure the card slots are no less than 16 lanes wide (x16) :



## FAQ

### ***I'm having multiple GPUs installed in my system. How to select my preferred GPU for full screen rendering?***

Right-click your system in the Project Explorer and select **"Set Preferred GPU Adapter"** from the context menu. Choose the GPU from the drop-down list. The selected adapter will get stored in your systems registry on a global level.

### ***The content seems to be transferred to my Windows hard drive. I need Vertex to transfer the content to my dedicated content hard drive with larger capacity and higher speed.***

#### ***How do I tell Vertex my content drives path?***

Vertex will store all projects and its assets at the path that has been defined as **"Default Project Path"** during the installation of Vertex. If later you need to change this, select your System from Project Explorer and search for **"Path"** by using the filter in the inspector window. Change either the **"Default Project Path"** or your project specific **"Local Project Path"** in order to change the drive or directory of transferred content.

Note: you might stumble upon the system property **"Content Drive"**. This value is only used for the calculation of **"Free Content Space"** (System Info).

### ***When manually seeking or spontaneously jumping to a cue in my sequence with a large number of tracks and clips, I see the clips being created sequentially with a short time delay. How to manually influence this interval?***

Select your **System** from your Project Explorer and navigate to **"Settings"** in the inspector window. Adjust the **"Render Element Create Interval"** (Default: 40ms) in order to speed-up or slow down this process.

### ***Why do I occasionally see jerks/judder in my playback, that look like missing frames or tiny jumps?***

Especially when dealing with different framerates (content framerates vs. sequence framerate vs. output framerate), small glitches in playback might become visible when frames are missing or need to be dropped in order to stay in sync.

Activate the parameter **"Frame Blending"** (default: false) for contents that require smoother rendering.

Consequently, frames will get interpolated. However, we do not recommend activating frame blending for any content in general, as this will have an impact on the performance of systems with a heavy load.

### ***My ASIO audio playback is occasionally cracking during playback. How do I solve this?***

It's highly recommended to use a dedicated hardware for audio playback. Ensure the system connected to your audio interface is not rendering any video output, handling DMX data nor is used as sync master of your session. The use of virtual ASIO devices such as Dante Virtual Soundcard is fully supported by Vertex. Please keep in mind these virtual devices are utilizing your systems CPU for their tasks while Vertex might claim the same resources. The same applies to USB connected audio interfaces.

We recommend using dedicated Yamaha or Focusrite PCIe cards as they come with a Datan chip onboard that is providing dedicated performance.

General advice:

- Make sure you've set your **"Asio Output Channel Count"** (Default: 0) to the amount of audio channels used in your project. You can find this parameter in your project settings.
- We also recommend **setting your ASIO drivers' buffer to above 1024 samples**.

### **My composition's background becomes visible for a few frames when seeking in other clips or using GotoCue/GotoTime as my system is creating the new clips too slow. How to tweak this behavior?**

In an ideal scenario the playhead is either running into your clips that have their **PreRoll** parameter set according to the content size (default: 2 sec), or the clips that you are jumping to are set to **"Pre-Loaded"** (default: false).

If both options can't be guaranteed there are further ways to enable a seamless jump:

- Temporarily pre-load your clips in case you know where to jump in advance. Trigger this script (e.g. via cue script code, ControlView button etc.) to pre-load any clips at a certain cue or time:
- **Playback1.TempPreloadCue [CueID]**
- **Playback1.TempPreloadTime [Timecode]**  
The clips will get unloaded automatically once jumped into them.
- define a **"Clip Hold Time"**. Your clip will still be rendered for the specified time after the playhead left the clip. Meanwhile the next clip can be loaded (either in foreground or background, depending on time and Z-hierarchy), while the previous clip is still present.

The **Clip Hold Time** can be defined for the whole sequence as **"Default Clip Hold Time"** (default: 50ms), or on single clip basis: Activate **"Enable Hold Mode"** (default: false) and specify the **"Release Hold Time"** (default: 1 sec) and – if desired – a **"Hold Fade Time"** (default: 13 frames). Once the clip specific **Hold Mode** is enabled, this value will overwrite the sequence's **"Default Clip Hold Time"**.

### **My playback is out of sync. How do I re-sync my playback automatically?**

Vertex is constantly improving its abilities to automatically synchronize the playback of multiple clips on multiple outputs and multiple systems. If you are experiencing playback out of sync, a very quick manual PAUSE/PLAY command will most likely solve your synchronization issue.

**Tip:** In most cases, out-of-sync playback is caused by the lack of your clips' pre-load time. E.g., skipping to a position in your sequence with clips that are not preloaded and instantly hitting PLAY will force all clips to start playing without being loaded. To counteract such events each sequence has its **"Load Goto Play Cue Wait Time"** property (default: 1 sec). When using the script **"GotoCue Play"**, this property defines a buffer in time for VERTEX to anticipate clips that are not set to pre-loaded or haven't been temporarily preloaded.

**My hardware's performance is absolutely not matching the content's requirements. Is there any chance – at least for pre-programming or rehearsal scenarios – to enable a fluid playback of the content?**

There are multiple options available, depending on your timeframe and emergency severity:

- transcode your content by using our build-in content transcoder to generate HAP files easily. Right-Click your content and select "**Transcode**" from the context menu.
- reduce your surface's resolution by using the surface property "**Down Scale Factor**" (Default: 1).
- reduce your individual contents framerate by using the content property "**Custom Video FPS**" (Default: 0).

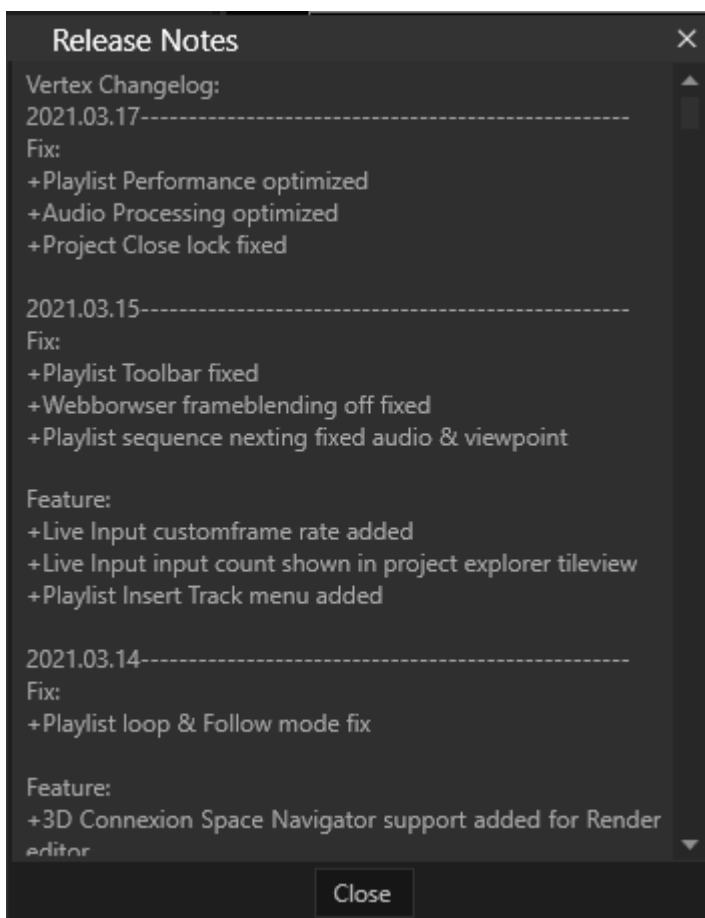
**I've set up my systems accordingly and applied all Windows and NVIDIA tweaks (all outputs w/ same framerate, mosaic, etc.) to get the best performance out of my hardware. But still Vertex is showing issues in playback behavior. What can I do?**

Some general adjustments can be made to ensure a higher performance:

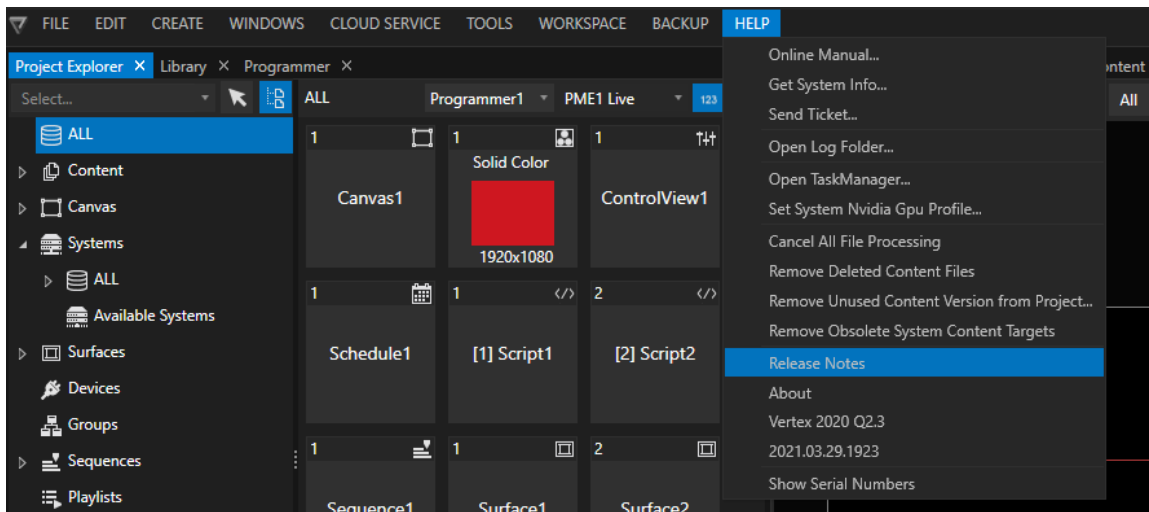
- set system property "**Disable Preview in Fullscreen**" (default: false) to **TRUE** to disable the UI's Render Editor Window in order to save resources.
- set system property "**Render LocalSystem Only**" (default: false) to ensure the particular system is only rendering its specific surfaces.
- Reduce your systems "**Render Window Size**" to the resolution that is actually required and limit your GPUs output quantity accordingly. Disable outputs that are not in use.

## 4.6 Release-Notes

- Each new VERTEX version has the **release notes** along with it
- With less clicks you are **able to check easily** what has **been fixed, improved** or **what is new**
- If you are **curious about our development**, the **changelog in the release notes** is always a good place to start



Where to find



- Just open **HELP** into the [main menu on top](#)
- Navigate to **"Release Notes"**
- A new window is opened

*Getting Started*



## 5 Getting Started

- *Start from Scratch and learn VERTEX step by step*
- *Or just read one topic to get background information for a specific task*

### Create your first project

1. [Learn how to create a project and which steps has to be done](#)
2. [Learn where to find global Project Settings](#)
3. [Basics about Project Load and Save](#)

### User Interface

[Learn more about the different Parts and Windows](#)

Get to know the most important and most useful editors. Get some useful background information about special settings

Learn Basics about User Interaction into VERTEX

### Add Content

[Different types of content in VERTEX and their specific settings](#)

### Manage Content

Manage your content into the Project Explorer

1. [Define User Properties like Notes, Color, Names](#)
2. [Sort content with Collections](#)
3. [Group Devices](#)
4. [Use the unique visioning feature and switch through Content versions with ease](#)

### Content Arrangement

Canvas, Clip Containers, Keyframes, Playback , Playlist, Video FX, .....

[Read the Introduction first to get the Basics](#)

Configure Outputs

Playback Live and/or in Preview

Enough Basics?

[Learn more about the Advanced Features](#)

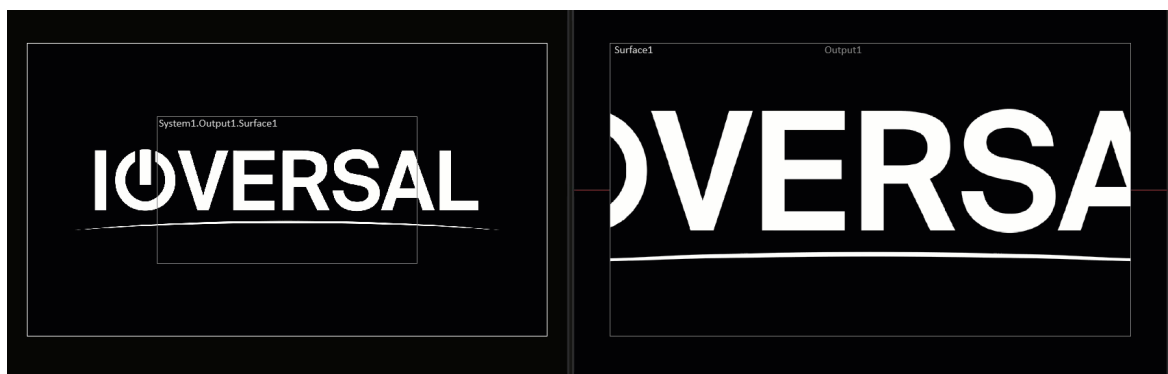
## 5.1 Canvas, Surface and Output

- Content arrangement, playback and rendering to output is basically done on three levels: **Canvas, Surface and Output**.
- Understanding the relationship and differences between these three levels is crucial.

### Workflow In 3 Quick Steps - From Canvas To Output:

1. [Import media](#) and [arrange it on a Canvas](#).
2. [Add surfaces to your canvas](#) and position them.
3. Assign [surface\(s\) to a system output](#).

## Surface Is The Link Between Canvas And Output



Surface Position in Canvas View

Output View

- The **Canvas** is VERTEX's virtual stage area where all media content is arranged.
- A **Surface** is a frame positioned onto the Canvas - much like a camera frame capturing a part of that stage.
- The **Surface** captures a 2D render texture with optional **warping & blending** and sends it to an **Output**.
- The **Output** plays out the rendered image to the connected hardware (screen, projector, LED wall, etc).

### Canvas Space



### Output Space



## Workflow and Advantages

For your convenience, every new VERTEX project starts with a Canvas in the size of your local Windows desktop. Canvas size can be changed anytime.

Per default, a Surface is automatically added and assigned to your output hardware in matching size and pixel resolution.

### Advantages

- Your content and **creative workspace is independent of your outputs.**
- You can always get started arranging your show without knowing the number of your total VERTEX Systems, their output routing, or the exact number of your total outputs.
- Your project can be can be rearranged on each of the three levels '**Canvas-Surface-Output**' at any time.
- This flexibility makes it easy to adjust your output hardware when your project grows in size and complexity.

## 5.2 Configure Outputs

- This Chapter guides you through your output configuration. Set up outputs, assign surfaces to outputs and configure settings for audio playback.

### Set up your Outputs

[arrange your systems output setup](#)

### Output Settings

[learn more about the output settings of a system](#)

### Surfaces

[the important role of surfaces and how to assign them to outputs](#)

### Softedge Blending

[information on tools for softedge blending](#)

### Audio Outputs

[how to set up audio devices and how to configure audio outputs](#)

### 5.2.1 Surface

- The Canvas Space **works separately** from the Output Space. A **Surface** is the **link between the two**.
- In order to play out content, Surfaces need to be **added to a Canvas** and **assigned to an Output** - either per drag-and-drop or from a context menu.
- Surface settings include **[Canvas Offset \(position of Surface on Canvas\)](#)** and **[Canvas Viewpoint](#)**, as well as settings for output positioning.
- **Multiple Surfaces** can be **assigned to the same Output** in split configurations.

## Surface links Canvas to Output

### Canvas Space



### Output Space

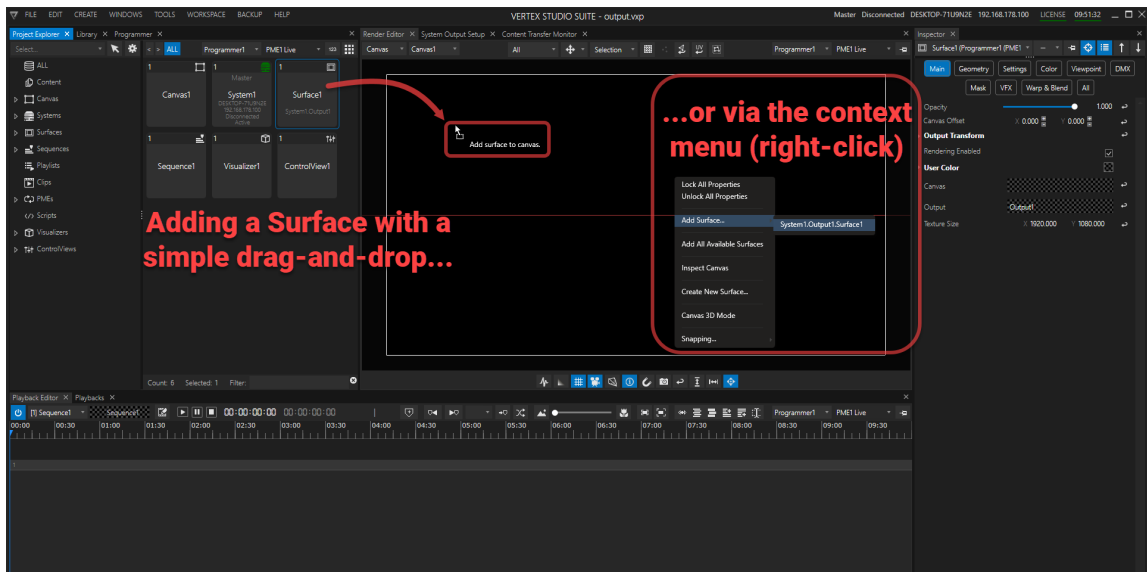


A Surface acts as a **bridge between Canvas Space and Output Space**. Like a camera frame capturing a **2-dimensional Render Texture** which is provided **for the Output**.

Because a Surface links both spaces, there are **transformational properties** in a Surface's Inspector **that work for Canvas and Output separately**.

Please also read the introduction to [Canvas, Surface and Output](#).

## Add / Remove a Surface



There are **two different options** for adding a surface to a canvas

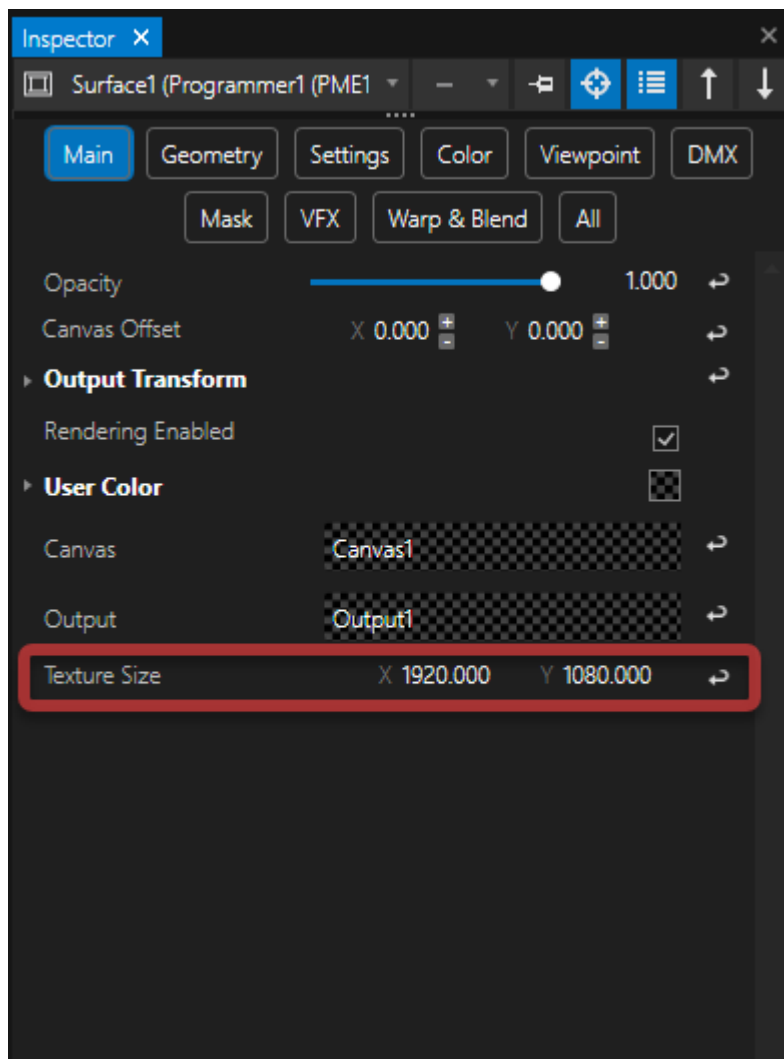
1. Per drag-and-drop from Project Explorer into Render Editor
2. or via the context menu (right-click in Render Editor) with the option of adding only one specific surface or all available surfaces at once.

## Removing a Surface from a Canvas

- Access the **REMOVE SURFACE** command via the context menu of a surface in the Render Editor. Choose between removing only one specific surface or remove all surfaces from canvas.

## Settings for a Surface

## Surface Size



The **Texture Size** defines the **size of the Surface**.

By default the texture size for new Surfaces is 1920 x 1080 Pixel

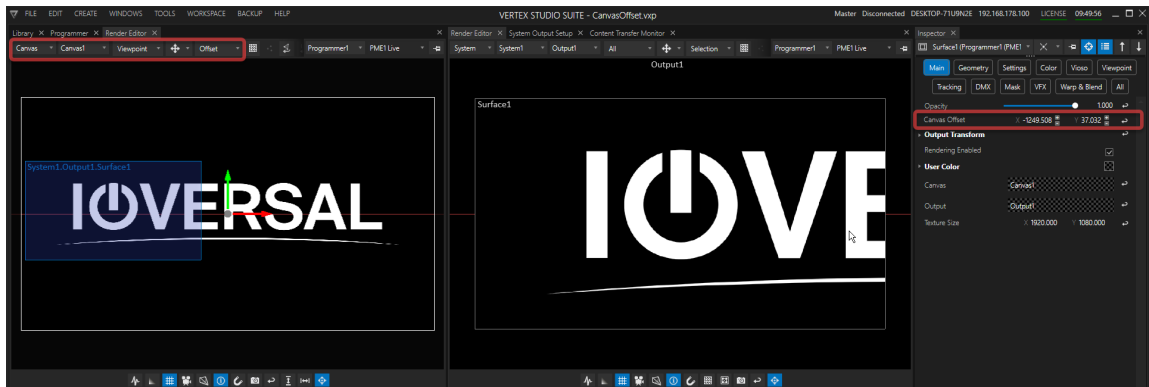




When adding a new system with outputs, VERTEX automatically creates a Surface for each new output.

In this case the Texture Size of the Surface matches the pixel size of the Output.

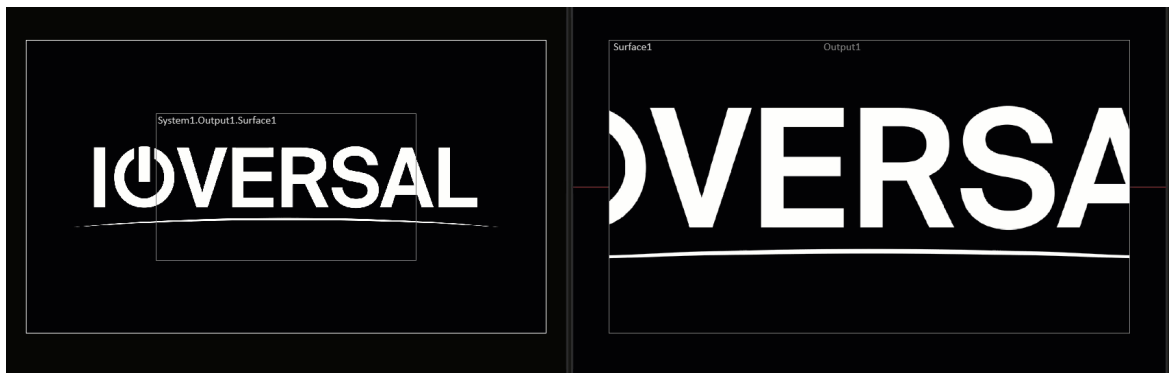
## Surface Position and Viewpoint on Canvas



The **position** of the Surface on a Canvas is called **Canvas Offset**.

The values for X and Y **offset** the frame of the Surface from its default position. Use them to reposition the Surface to its desired spot.

Switch to **Canvas View** in the **Render Editor** to see what you set:



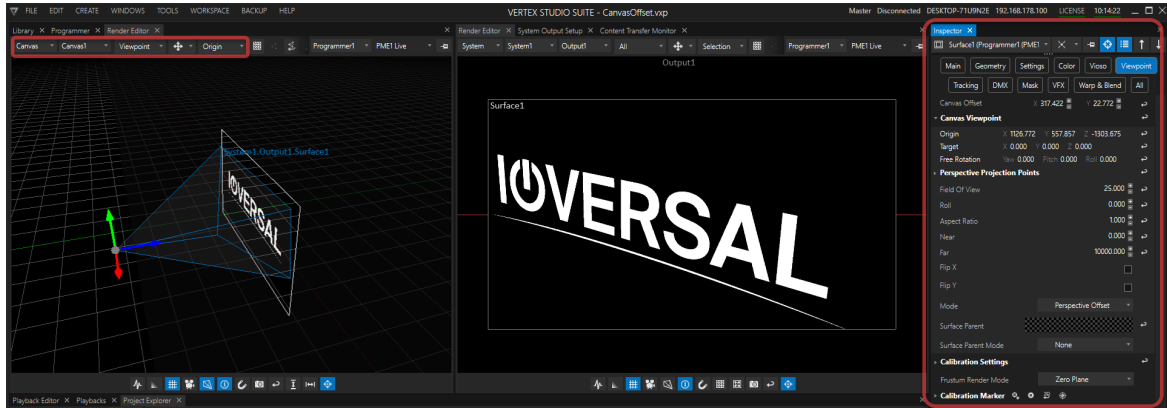
Surface Position in Canvas View

Output View

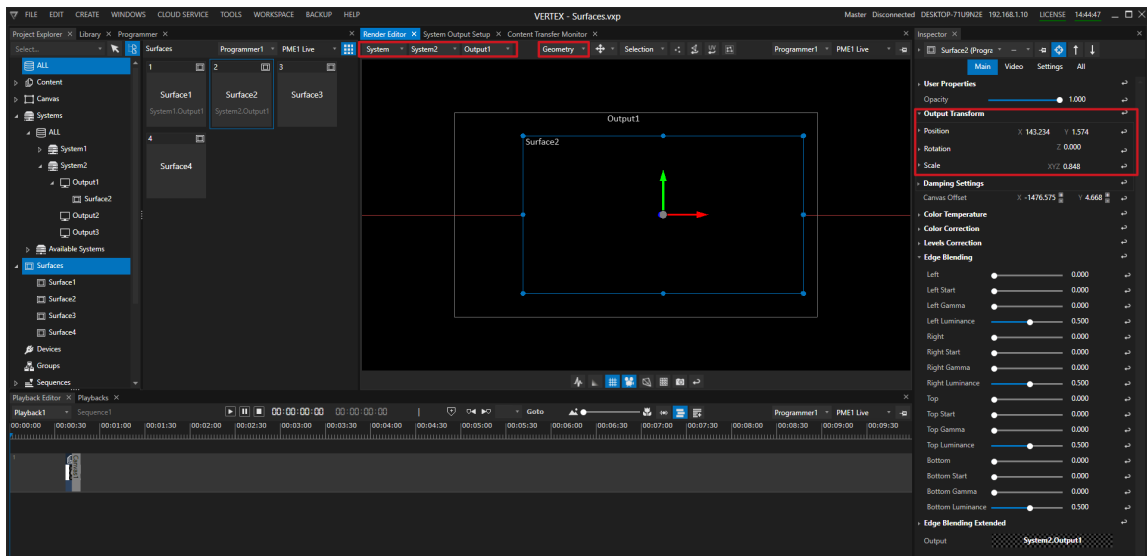
## Canvas Viewpoint

The **Viewpoint Tab** of a **Surface's Inspector** contains various settings to **define** and **manipulate** the **Surface's viewing angle** onto the **Canvas** for **specific 3D views**.

In most cases the default settings should work. VERTEX automatically calculates the viewpoint origin based on Texture Size and Canvas Size.



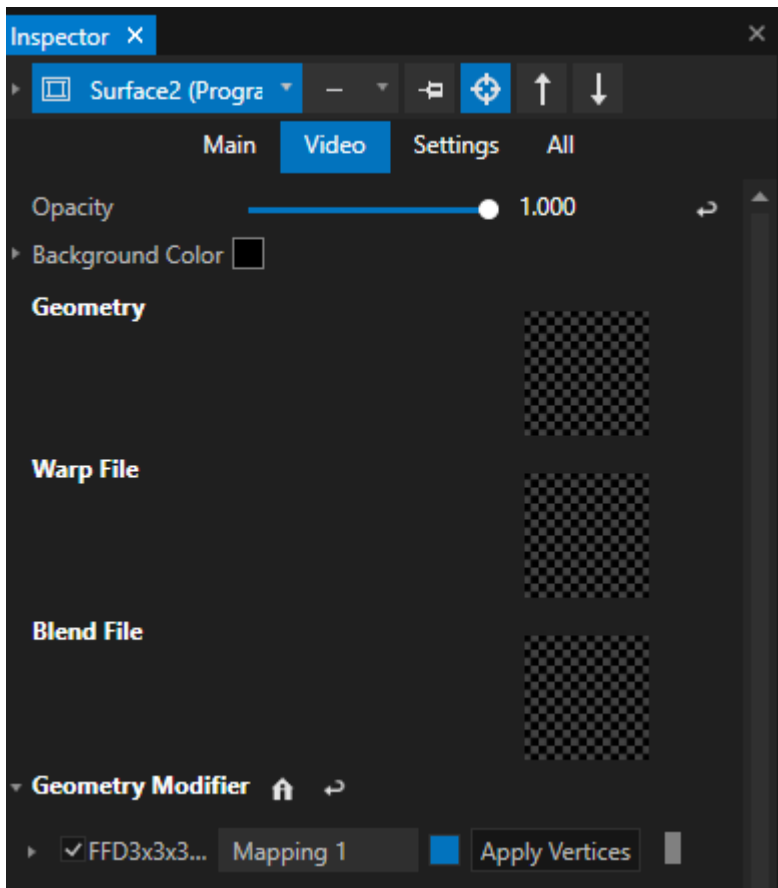
## Output Position



The **Output Transform Settings** adjust the position of the Surface in its output .

- Select view **System** in the Render Editor and **select the Output** that is assigned to the Surface
- Switch to **Geometry Selection** to select your Surface
- **Transform your Surface** by setting the values in the Inspector or move the Surface with your mouse in the Render Editor.

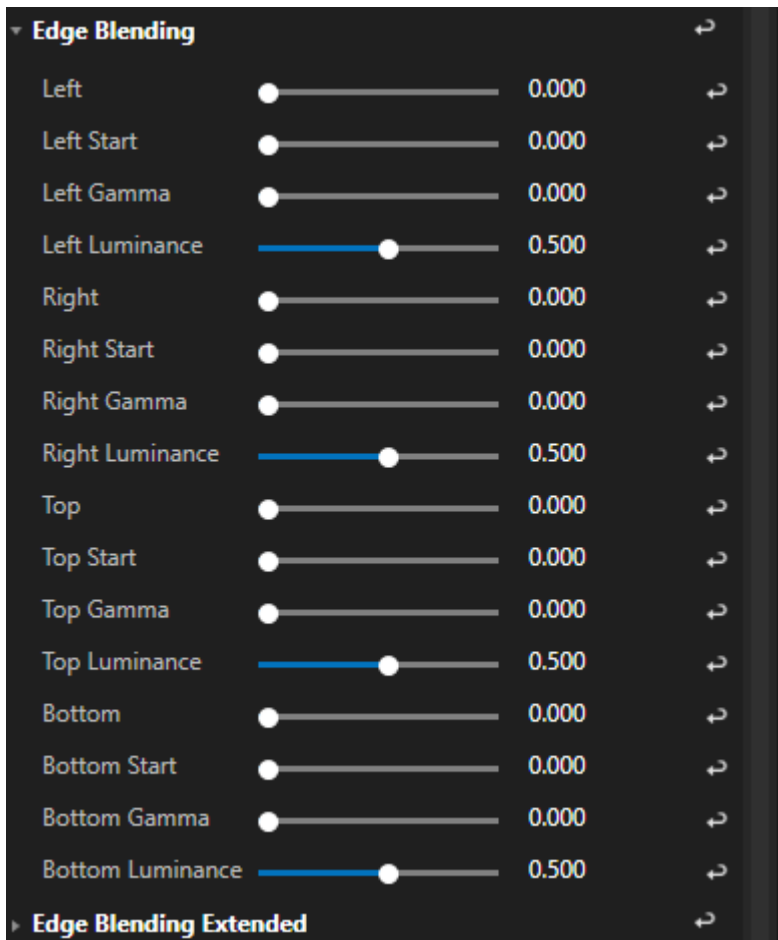
## Geometry Modifier and Warp Files



Each Surface has Property fields for

- A **3d Geometry** that can assigned by drag and drop from Project Explorer
- A **Warp file** (Currently only VIOSO Warp Files supported)
- A **Blend file** (Currently only VIOSO Blend Files supported)
- [Geometry Modifiers](#) to do a Warping

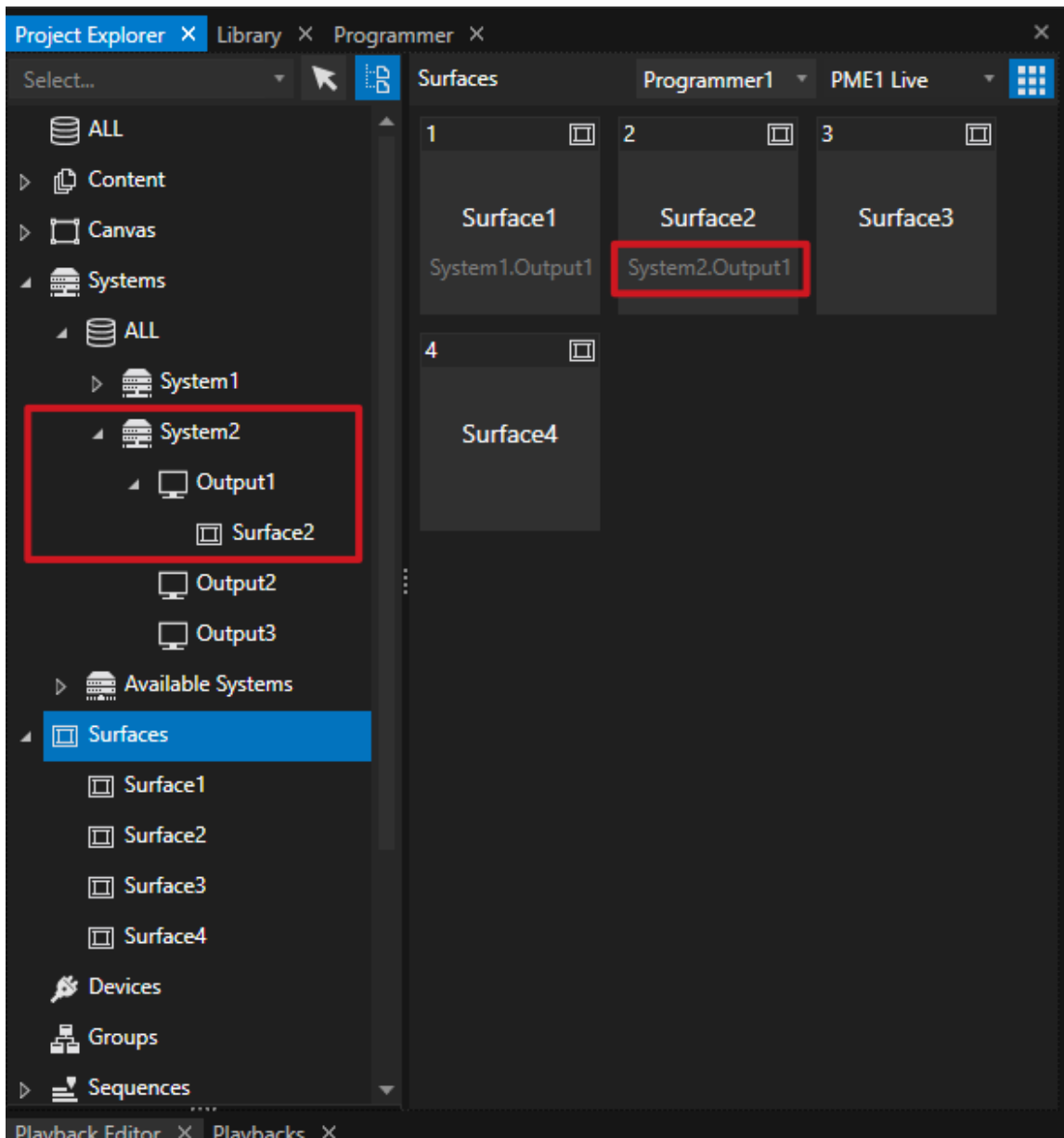
## Edge Blending



*Each Surface has Edge Blending Parameters to set up an Edge Blending*

## Assign Surface to an Output / Remove Surface from an Output

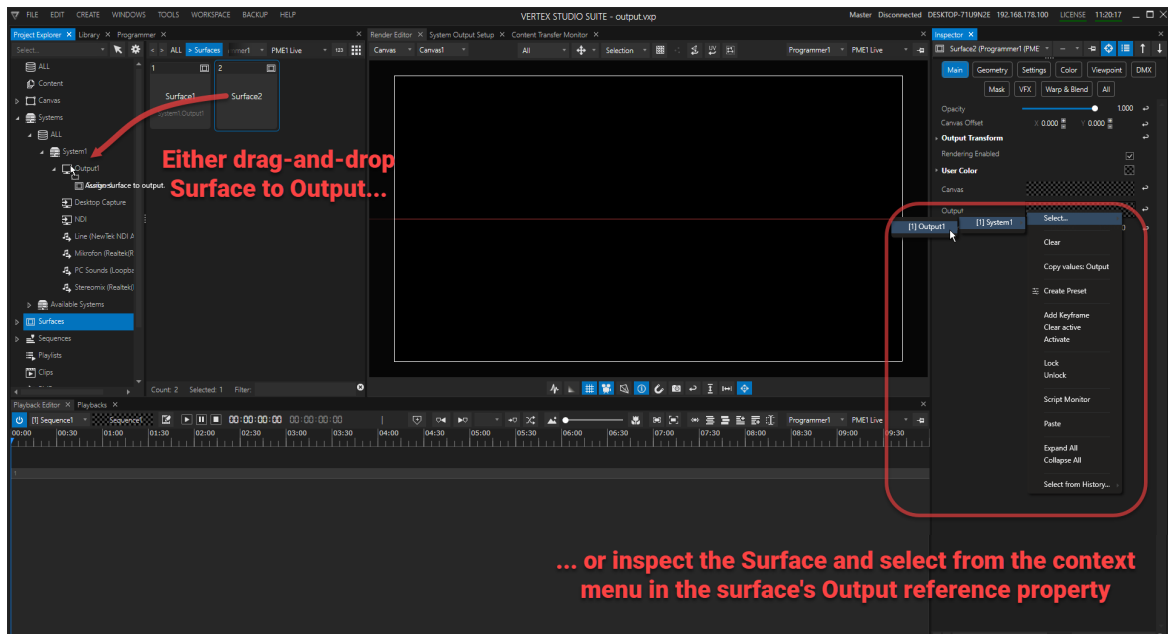
### Output Reference



The output reference for each Surface assigned to an Output is shown in various places:

- the **Surface tile** in the Project Explorer
- top-left corner of the **Render Editor** (i.e. System2.Output1.Surface2)
- in the **Inspector** of a Surface under the tabs Main and Settings.

## Assign Surface to an Output



There are 3 ways to assign an already existing Surface to an Output:

1. **Drag-and drop the Surface tile** from Project Explorer to a system's output in the tree-view
2. or drag and drop the Output from the tree-view of Project Explorer to the Surface's Output reference property in the Inspector,
2. or **access the context menu** (right-click) in the **Surface's Output reference** property in the Inspector and select your desired output from there.

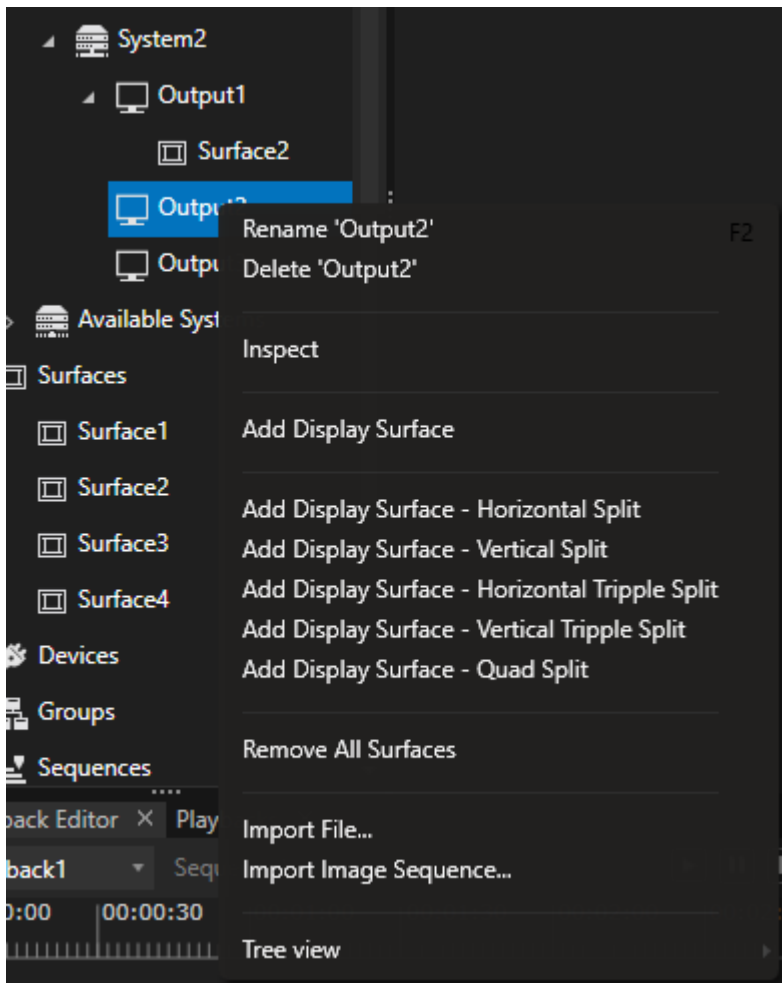


When starting a new project or add a System as session member to your project, for every output a Surface is automatically created.  
This surface already is assigned to the corresponding output.

**If no Surface exist or if you want to add more than one Surface to an output:**

Select your output from Project Explorer or System Output Setup and access the context menu.

The texture sizes are automatically set for each new Surface.



## Remove Surface from an Output

There are **3 ways to remove a Surface from an Output**:

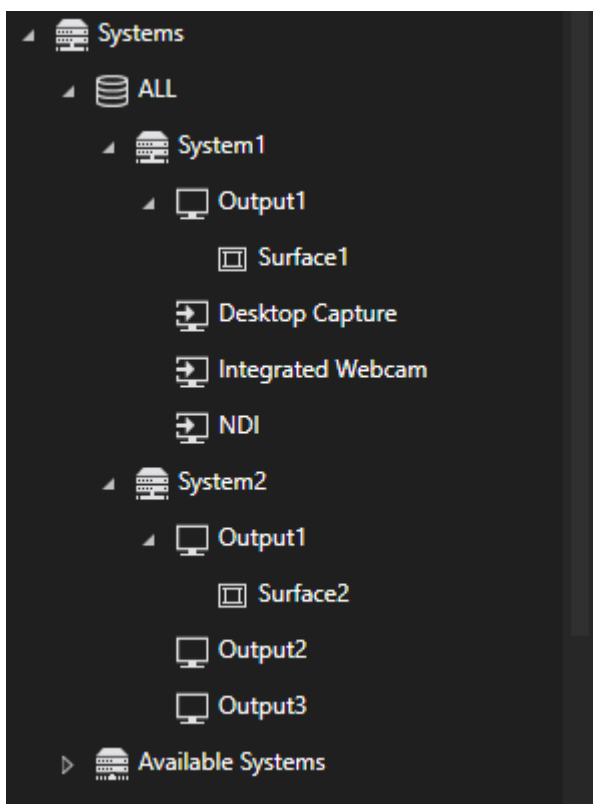
- Go to **Project Explorer** and navigate to your Surface, access the context menu and select **Remove Output Reference**.
- **OR**
- Delete the Surface as Child Element from an Output.
- **Surface Inspector**: Main Tab, go to **Output Reference** and select **"Clear"** from the context menu (right-click).

## 5.2.2 Output

- Outputs are **closely connected with the term Surface**. Without having a **Surface assigned** there will be only black
- Outputs **can be previewed into** System View of the **Render Editor**
- There are **additional Settings** for each Output into **Inspector** - The **System Output Setup** helps you with setup

## Output and Surface

- **Surfaces** connect Canvas Space and Output Space
- To **playout content** into an Output, you have to assign in minimum 1 Surface to this Output.
- Read more about [the basics behind](#)
- Read more about a Surface and [how to assign it to your Output](#)



Outputs of a System and the assigned Surfaces into Project Explorer



## System Output Setup

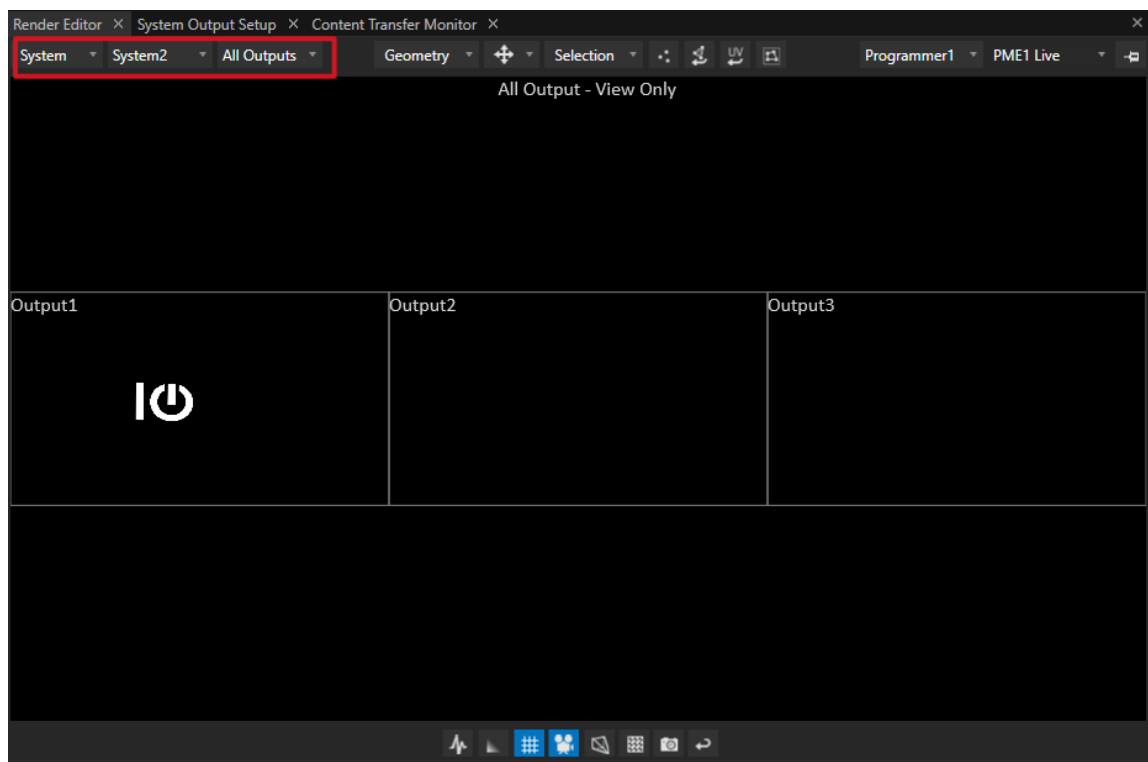
- The [System Output Setup Editor](#) supports you to setup the Outputs of your System

## Preview Outputs into Render Editor

The **System View** into the **Render Editor** gives you options to

- **Preview all Outputs** of a System
- **Edit a specific Output** and the assigned **Surfaces** and do e.g. a **Warping** on this Output

### All Output View

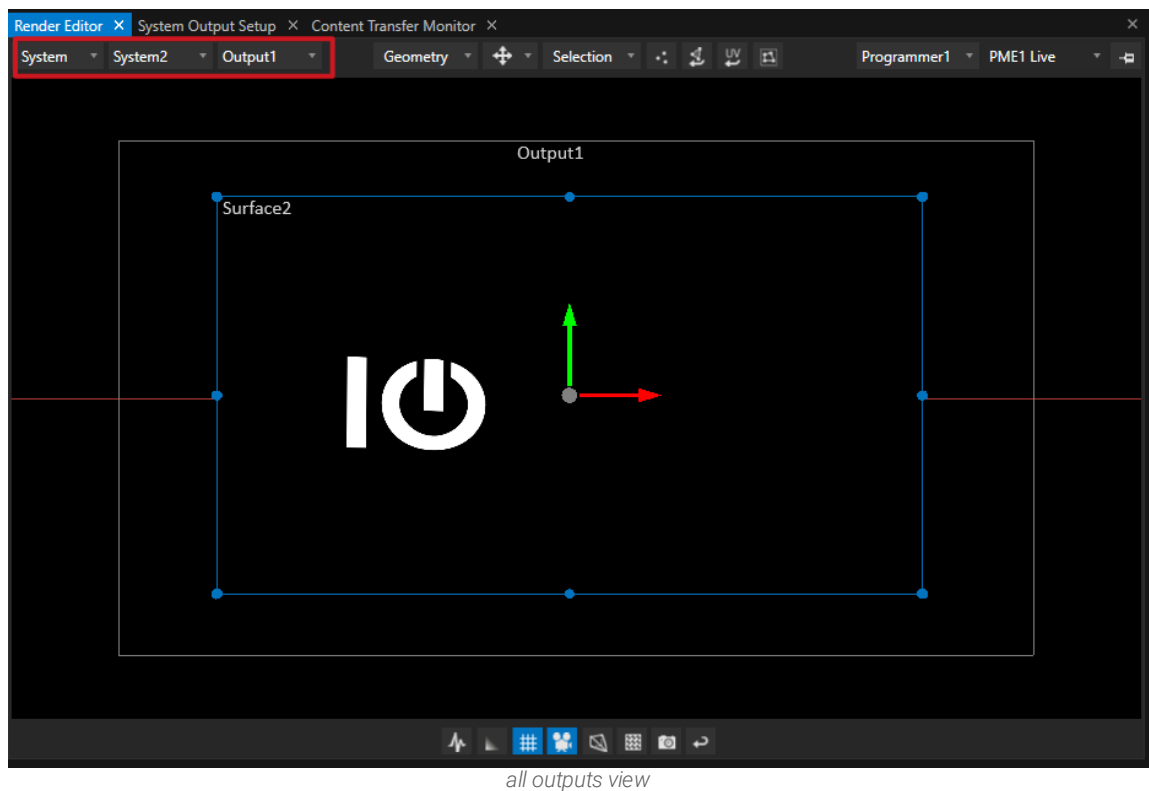


all outputs view

*All Outputs into Render Editor previews you the Render Results for all Outputs of your System. This is View only - Switch to a specific Output to edit this Output*

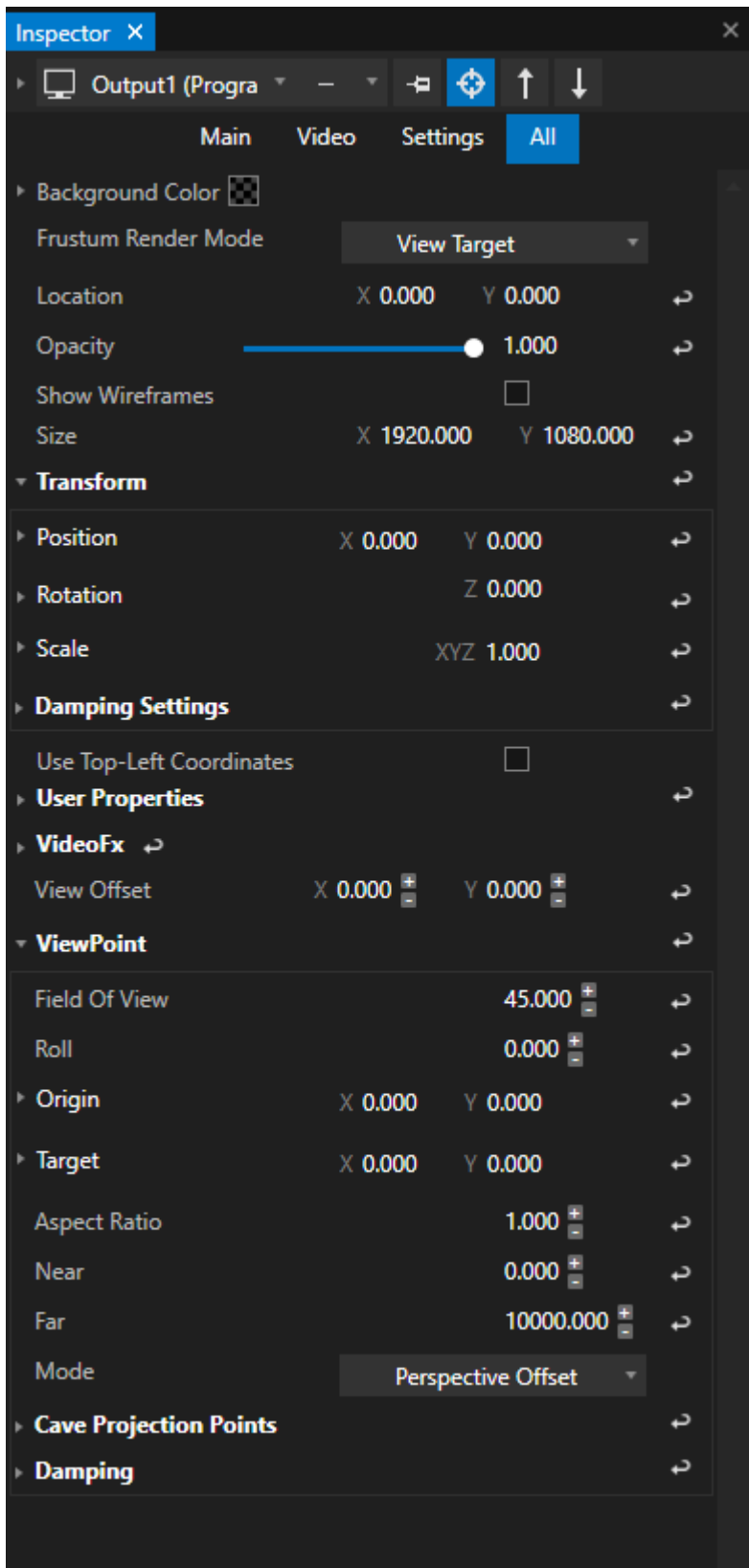
*You also are able to switch through other Systems into your Project if working into a Session and preview Outputs*

## Output View



*Into Systems View of the Render Editor you are able to choose a specific Output of your System. You have access to all Surfaces that are assigned to this Output: Edit them, so a Warping, modify Vertices*

## Settings



Settings for an Output

As for every element into VERTEX, also Outputs offers you a bunch of Settings and Options

As **regular user**, you mostly need only the **Basic Settings**:

- Position ( of the Output)
- Size (of the Output)
- Show Wireframes
- Opacity

VERTEX supports you with default Values. When you did System Output Setup before, you usually don't have to change Output Settings....

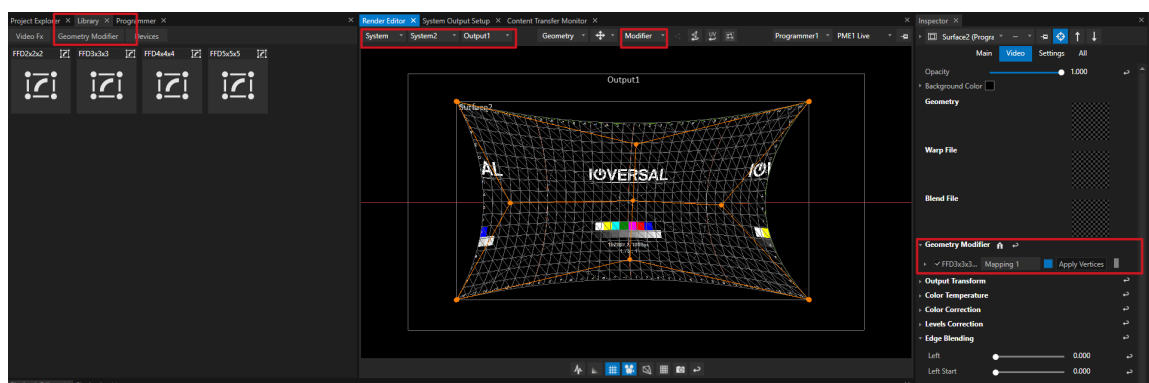
**For Edge Cases and Complex Setups, there are Advanced Settings available for each Output:**

- Transform Output,
- Change Viewpoint of the Output

## 5.2.3 Warping

- **Warping for an Output is done on a Surface**
- Open the **Render Editor**, switch to **System View** and select an **Output** to edit.
- **Combine different Freeform Deformers on a Surface** or even modify single vertices.

## Workflow



- **Open Library**, select **Geometry Modifier**
- Go to **Render Editor**, Switch to **System View**, Select your System and an **Output**
- **Select a Surface** that is assigned to your Output

- Drag a FFD from Library to a Surface

or

- select single Vertices of your Surface
- Drag an FFD to this Vertex



Warpings also could be done in the same way for a Clip Container.

In this case the Warping is only valid for one single Clip Container and not related to the whole Output

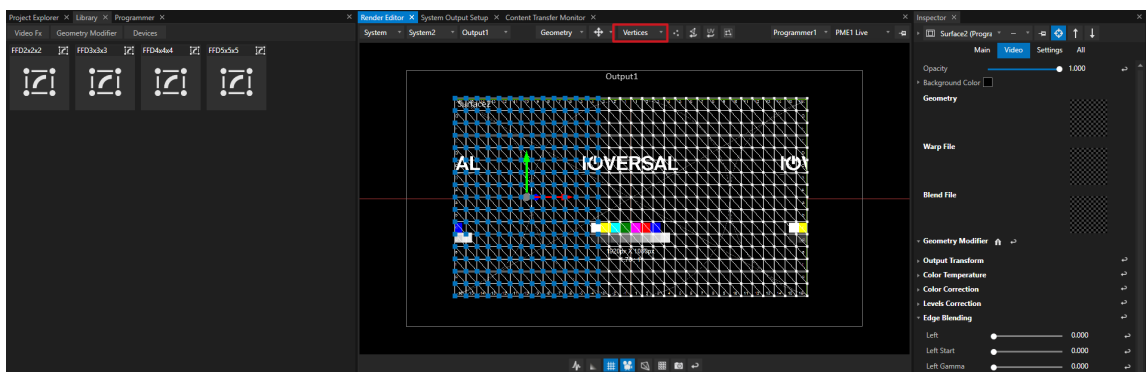
## Select Vertices and add FFDs

VERTEX gives you the option to **add several FFDs on different parts of your Surface** to make a more detailed warping.

Please follow the steps below:

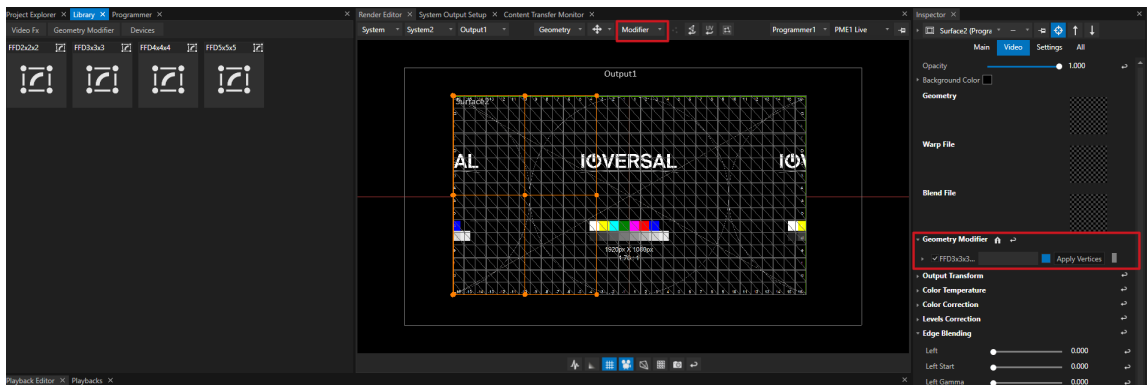
### Select single Vertices

Switch Selection Mode of Render Editor to "Vertices"



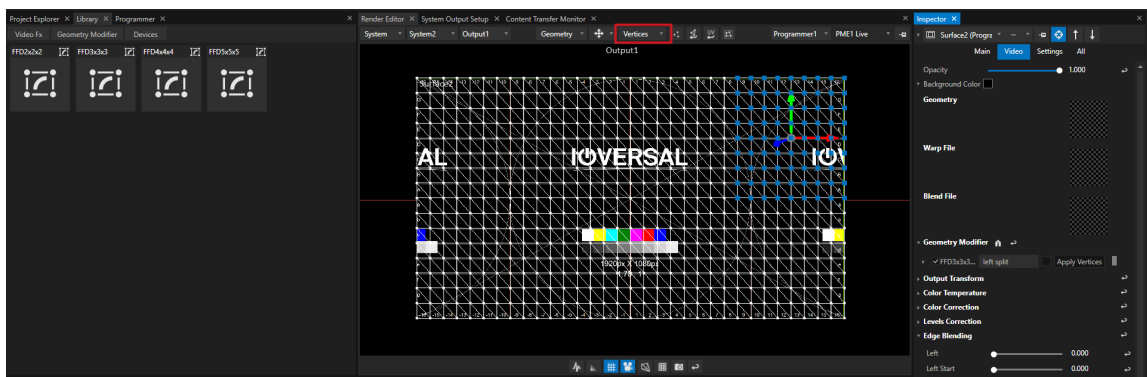
### Drag an FFD from Library to these Vertices

or use Apply Vertices Button in Inspector to apply the selected Vertices to an already existing FFD



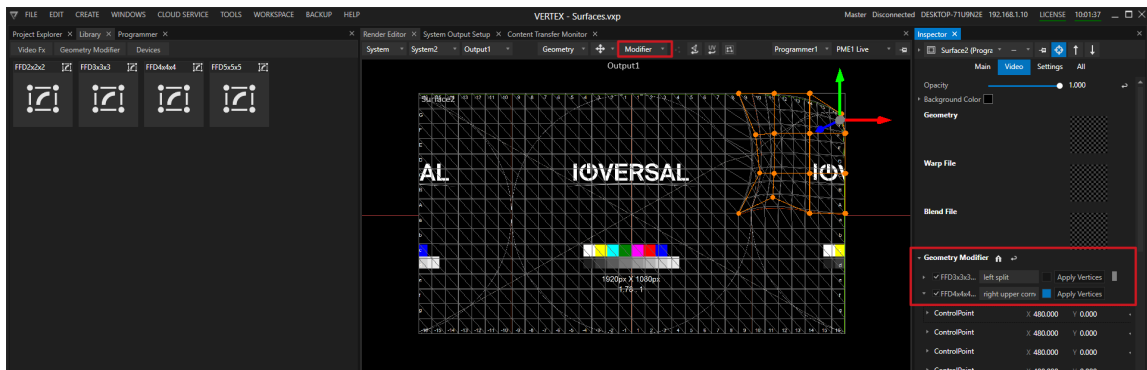
## Do another selection of Vertices

Switch Selection Mode of Render Editor to "Vertices" before



## Drag an FFD from Library to this Vertices

or use Apply Vertices Button into Inspector to apply the selected Vertices to an already existing FFD



### Select an FFD to show into Render Editor

When working with multiple FFDs on a Surface, you are able to select one of your FFDs with the blue Button (next to each FFD) into the Geometry Modifier Section of the Inspector

### Shortcuts for Render Editor

- Use **arrow keys to jump** to next FFD Control Point
- Hold **Shift** and use **arrow keys** to change **Control Point position**
- Hold **CTRL** and use **arrow keys** to jump into the middle of 2 control points and **select them both to modify**

## VIOSO AutoCal

With VIOSO Autocal, Warping could be done camera-based.

Every Surface also has property fields where you can add VIOSO Blend and Warp Files



*ioversal still works on the integration of VIOSOs Autocal Tool into VERTEX*

**The VIOSO feature set will be available soon.**

Warp and Blend files could not be processed with the current VERTEX release version

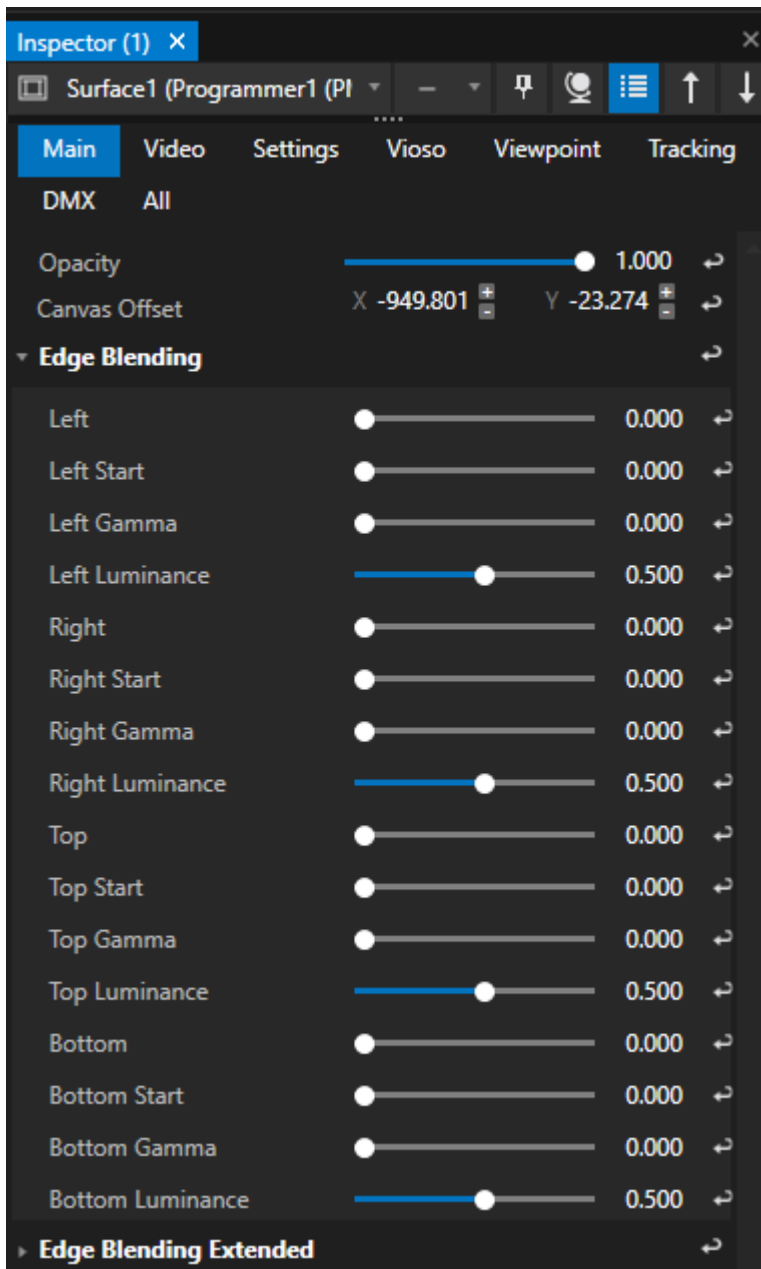
Contact us for further questions on this: [support@ioversal.com](mailto:support@ioversal.com)

## 5.2.4 Softedge Blending

- Every Surface has a **set of edge blending parameters** to create a softedge blending
- **The QuickBlend Wizard** supports you to create fast and easy blendings based on ioversal's blending algorithms
- Use **VIOSO Calibrator** for a camera-based projector and softedge calculation

## Edge Blending Parameters for a Surface





*Inspector with the Edge Blending Parameters of a Surface.  
Advanced Edge Blending Parameters extend to blending options for edge cases.*

Every **surface** offers you a **set of parameters for edge blending**.

With **4 parameters for each side**, VERTEX gives you many options to create a blending.

**Use Edge Blending Extended parameters** to adjust a blending of a large projection which is only partially overlapped by a smaller projection.

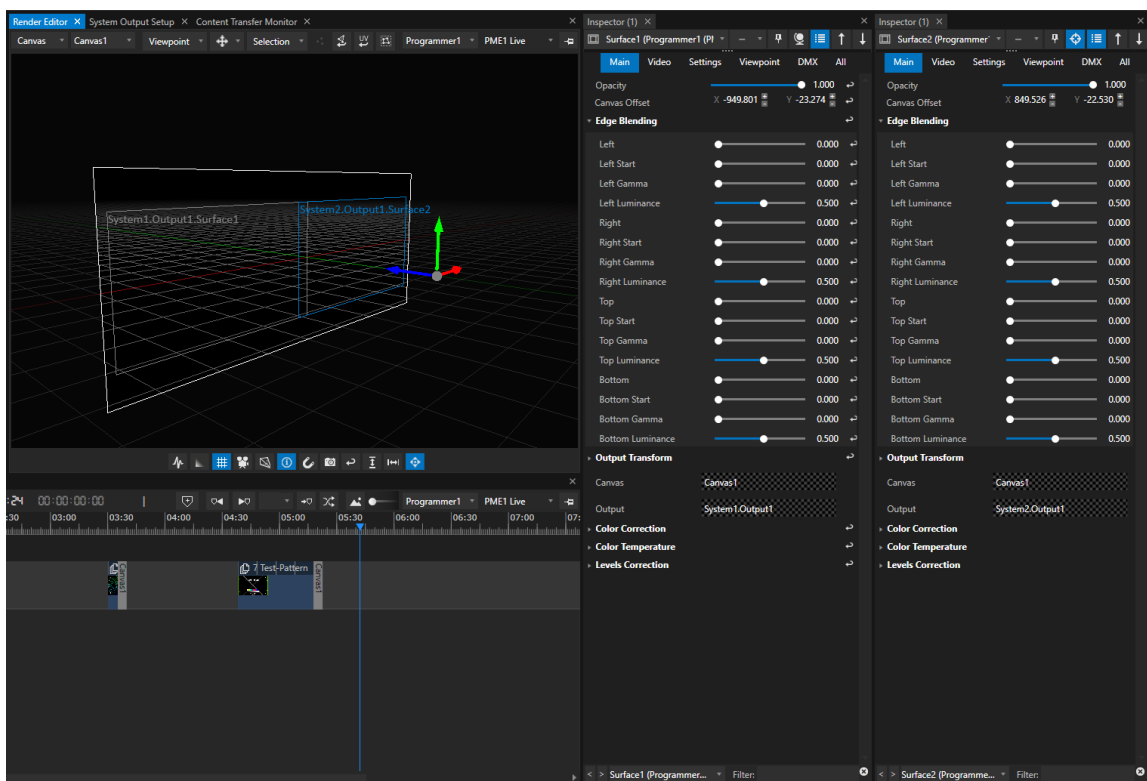


### Work with a second Inspector

For a soft edge with two projectors, open a second inspector. Dock the window next to the first inspector.

Select the first surface in first inspector and use the "Pin" button to fix the first surface to the inspector.

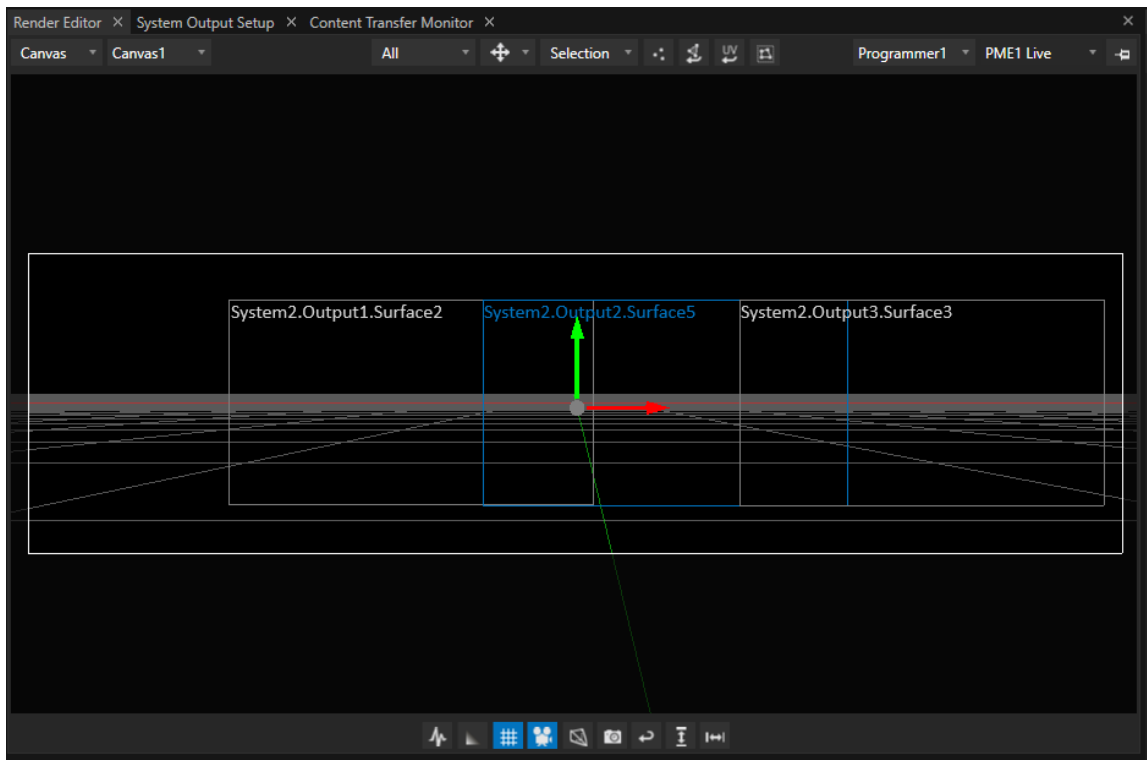
Select the second surface in the second inspector. Open Edge Blending Parameters for both.



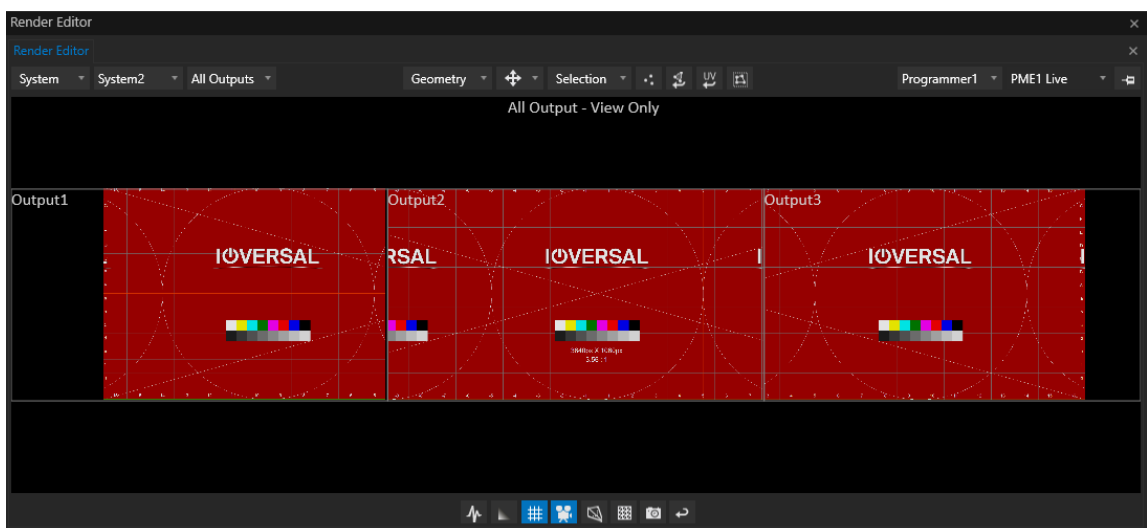
Workspace example for an edge blending:  
2 surfaces on a canvas, each surface is pinned to an inspector.

## QuickBlend

- The Quick Blend wizard **helps you to create edge blending for groups of two surfaces or more**
- If the surfaces **overlap** is set, the wizard **calculates the blending automatically and sets the corresponding edge blending parameters for all surfaces**
- Afterwards you only have to bring your Blending to perfection and do **fine adjustment**



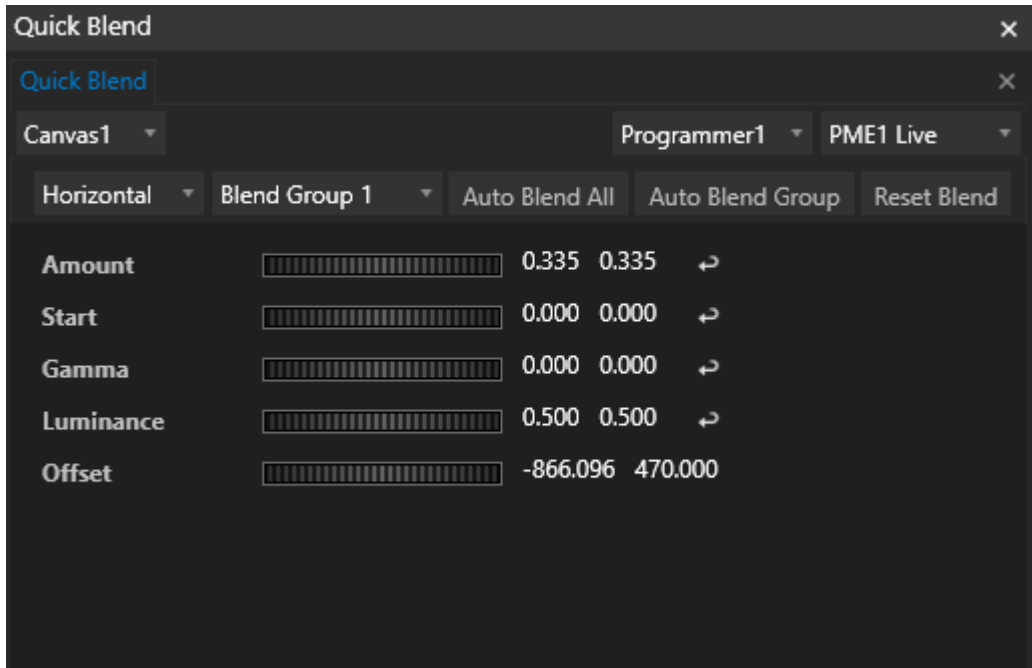
Example set-up: three surfaces on a canvas - each overlaps the next one.  
Final result should be an edge blending for three projectors to have a continuous single projection.



initial situation: outputs without blending

1. Go to main menu, select "Window" and open a new Quick Blend editor

## 2. Select your canvas in the Quick Blend editor



## 3. Select if you want to combine a horizontal or a vertical blend group

*If there are multiple surfaces on your canvas, QuickBlend creates groups for every two neighboring surfaces.*

## 4. Adjust the canvas offset for surfaces to finally define the overlap.

*By default the overlap that was set before for each surface is taken into account.*

*As a result of the fine adjustment the overlap should correspond to the overlap of the projectors.*

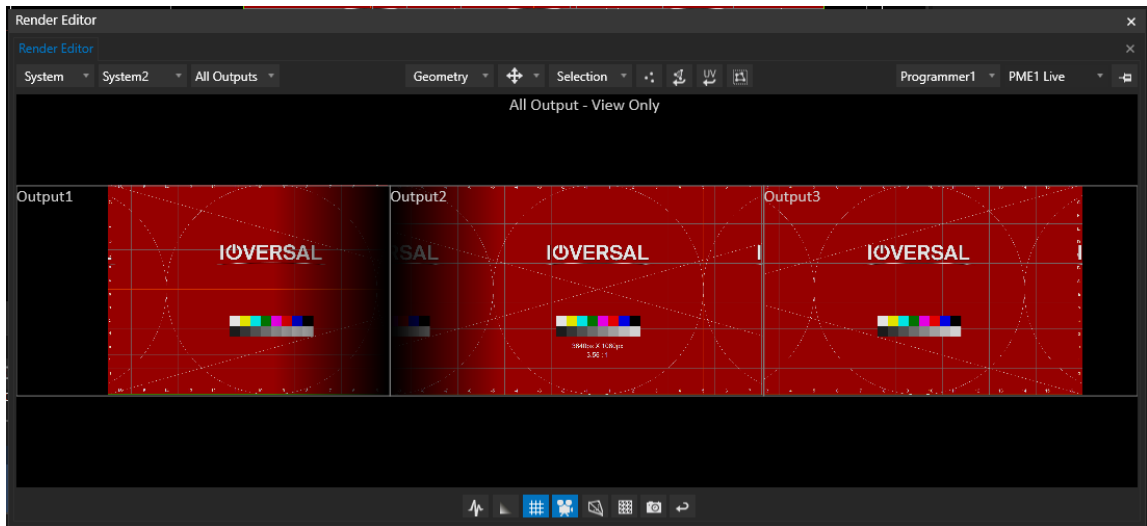
*Use the **slider** to influence the **canvas offset of all surfaces** of your group.*

*Use **first value field** to set the **offset of the first surface** of your group.*

*Use **second value field** to set **the offset of the second surface** of your group.*

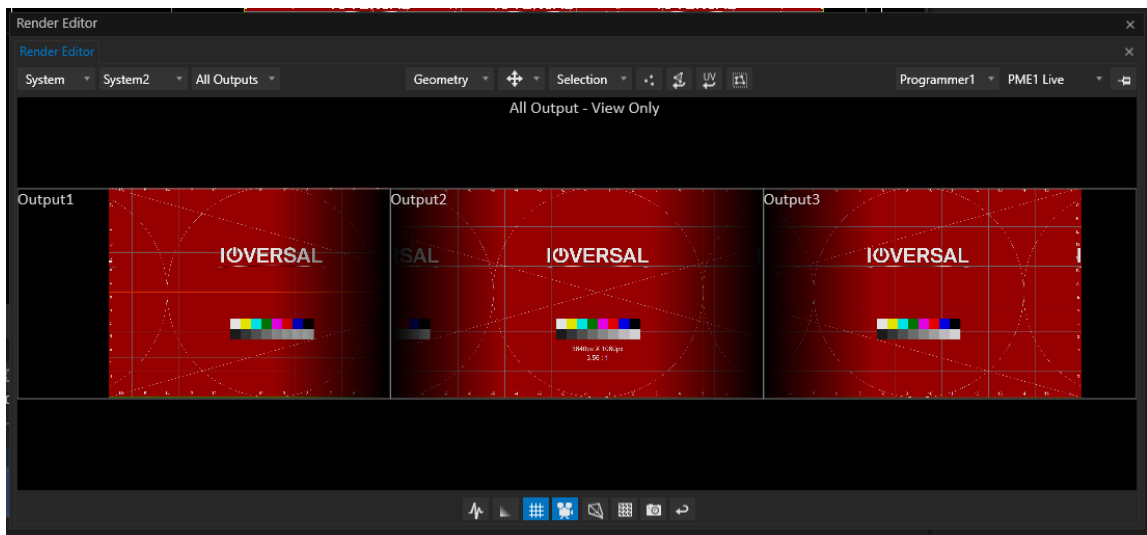
## 5. Select "Auto Blend Group" to blend a group

*Repeat the steps for all available horizontal groups*



*Blending for the first group of two horizontal surfaces*

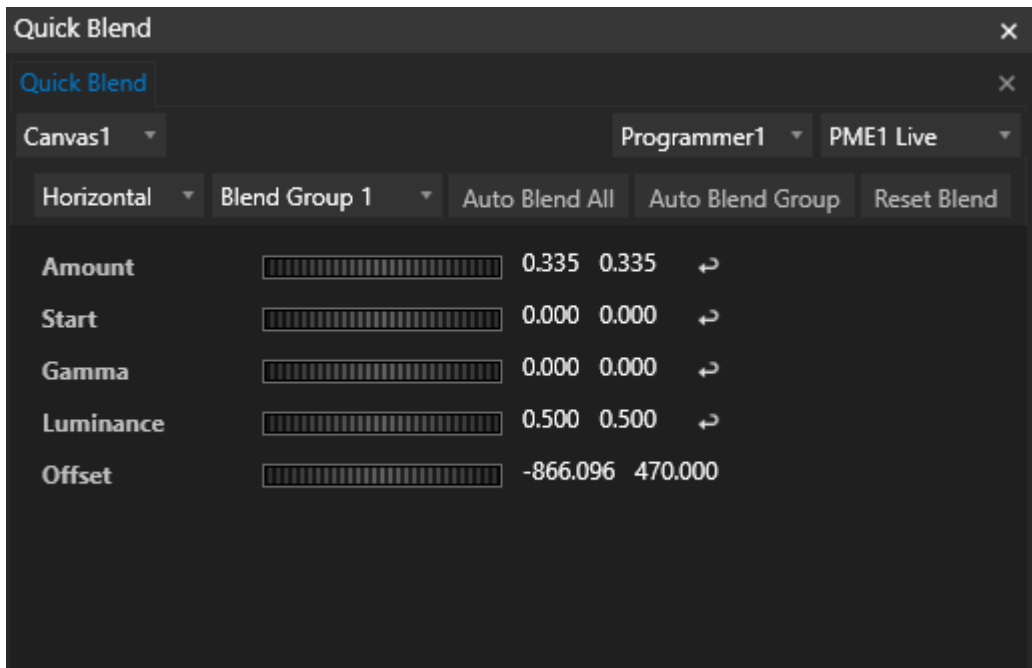
## 6. Select "Auto Blend All" to blend all available surfaces



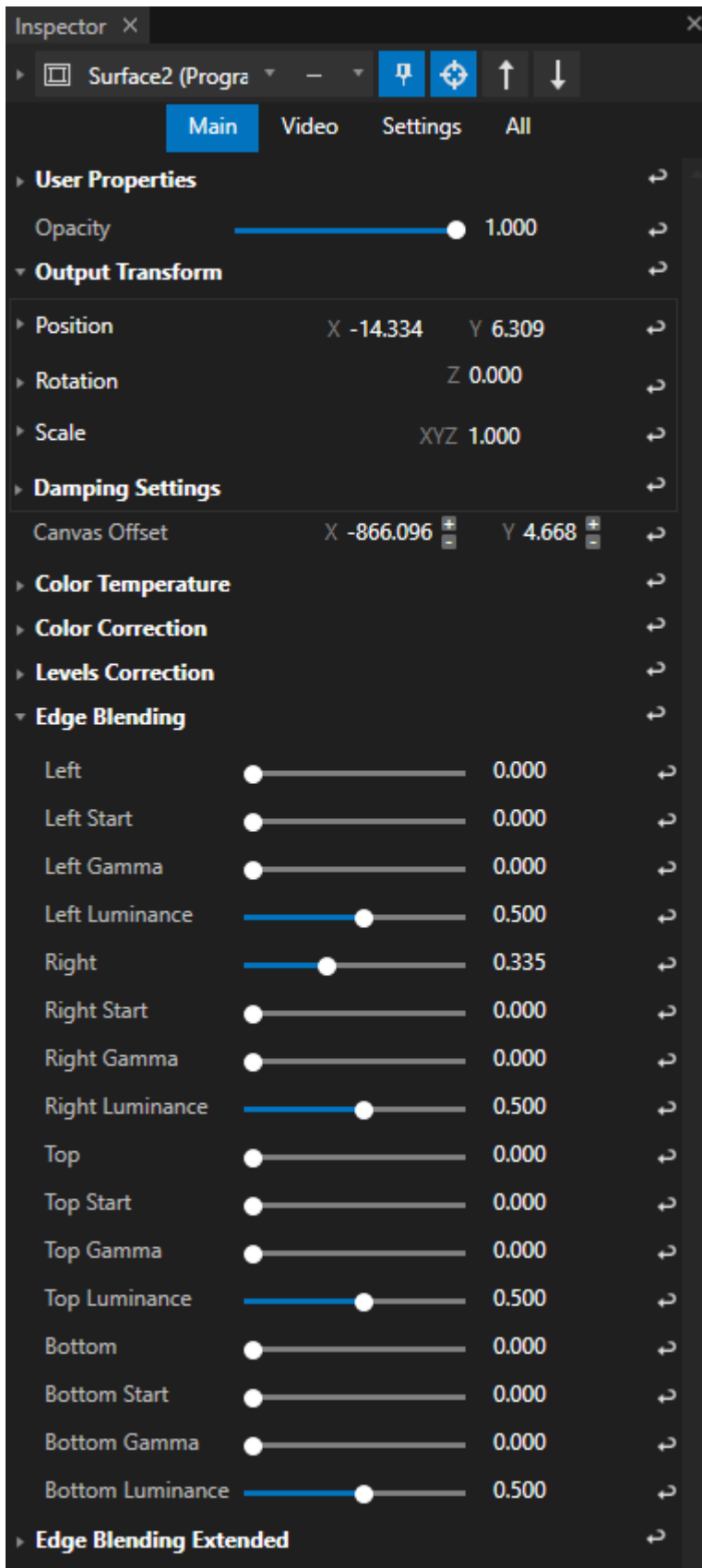
*Blending for all horizontal surfaces*

## 7. Do fine adjustments for blending with the help of Quick Blend parameters

*Fine tune your blending with parameters for amount, start, gamma and luminance*



Result: for every selected surface the edge-blending parameters are set by Quick Blend



*QuickBlend sets the Edge Blending Parameters for each Surface*

## 5.2.5 Audio Outputs

- VERTEX lets you select between **preview audio** i.e. for programming and **live audio** for the final playout.
- You can **assign preview and live audio to your ASIO or standard audio cards or interfaces**. You can **switch the audio clock** between standard audio and ASIO.
- **For each system you can set Windows volume, live and preview volume.**

## Preview and Live Audio

There is a main difference between **preview audio** and **live audio**:

### Preview Audio

- Preview audio plays out all audio from all playback mixing engines - That includes all playbacks in PME live and all playbacks in one (or maybe more) preview PMEs.
- Preview audio is played out from every system in your project.

### Live Audio

- Live audio plays out only audio from playbacks that are running in PME Live.
- Live audio is played out by an defined audio system that could be set for a canvas or a whole system.

By default live audio is played out for all canvas on the same system. It is possible to define a certain audio system for a canvas.

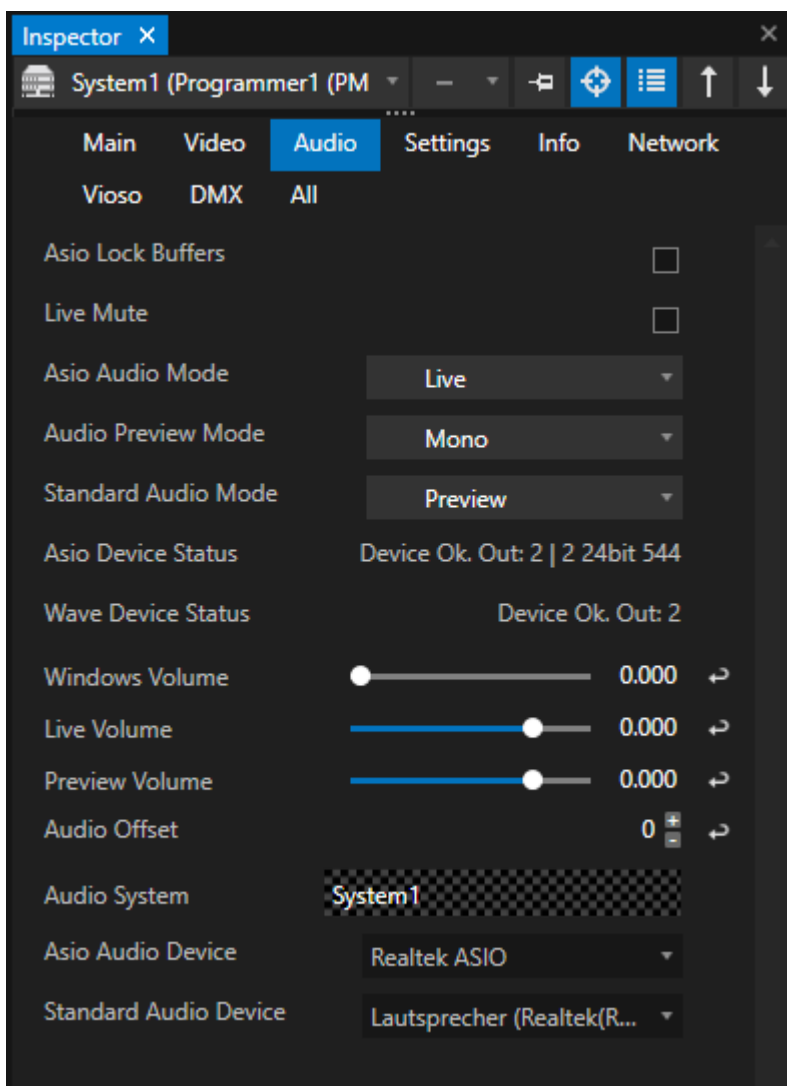
## Workflow for Audio Output Setup

1. select a standard audio device for your system
2. select an ASIO audio device, if connected to your system
3. define audio devices for live audio and for preview audio

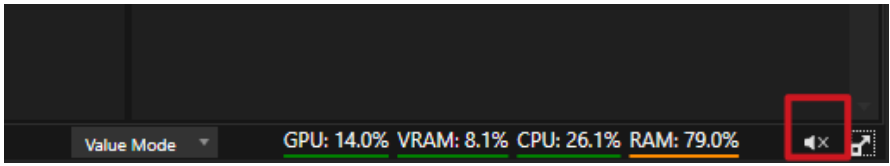


4. check sync-clock settings (sync clock is changed when live audio device is set)
5. adjust volume settings for your system
6. do final adjustments like audio offset or advanced settings

## Audio Settings for a System



<b>ASIO Lock Buffers</b>	<i>Default: disabled</i> <i>Enable recommended if you use an USB audio device.</i>
------------------------------	---

<b>Live Mute</b>	<p>Default: disabled</p> <p>When enabled, live audio output is muted.</p>
<b>ASIO Audio Mode</b>	<p>Select whether the ASIO audio device should be used for Preview Audio or Live Audio.</p> <p>The difference between the both is explained above.</p> <p>For high quality live output, we recommend an ASIO audio device.</p>
<b>Audio Preview Mode</b>	<p>Choose between a monoaural fold-down, a stereo mix of your preview listening or just keep the channel mapping one-to-one.</p>
<b>Standard Audio Mode</b>	<p>Select whether the standard audio device should be used for Preview Audio or Live Audio.</p> <p>The difference between the both is explained above.</p> <p>For high quality live output, we recommend an ASIO audio device.</p>
<b>ASIO Device Status</b>	<p>Shows the status of the ASIO audio device you have selected.</p> <p>Shows the number of output channels that are being used by Vertex, the sampling rate that is detected and the total number of outputs channels of the interface.</p> <p>For DANTE cards with a high number of channels: reduce channels in Project Settings to ensure a better audio playback performance.</p>
<b>Wave Device Status</b>	<p>Shows the status of the selected Standard Audio Device.</p> <p>Shows the number of output channels that were detected by VERTEX.</p>
<b>Windows Volume</b>	<p>Sets Windows system volume for the selected VERTEX system.</p>
<b>Live Volume</b>	<p>Sets the master volume level of the live audio output for this system.</p>
<b>Preview Volume</b>	<p>Sets the preview audio volume level for this system.</p> <p>For <b>Preview Audio Mute</b>, please use the mute button in the status bar at the bottom of the UI.</p> 
<b>Audio Offset</b>	<p>Sets a global Audio offset for this system, for example to adjust synch issues in audio and video playback.</p> <p>Values are set in milliseconds.</p> <p>Negative Values: Audio is sent out later.</p> <p>Positive Values: Audio is sent out earlier.</p>

<b>Audio System</b>	<p>Defines on which VERTEX system live audio should be played out.</p> <p>By default and when working only with a local system without session members, this is set always to the same System.</p> <p>Switch the system in backup scenarios where another VERTEX system should take over the audio payout.</p>
<b>ASIO Audio Device</b>	<p>Select an ASIO interface from a list of all available interfaces that are connected to your system.</p>
<b>Wave Audio Device</b>	<p>Select the standard audio device from a list of available Windows audio devices in your system.</p>



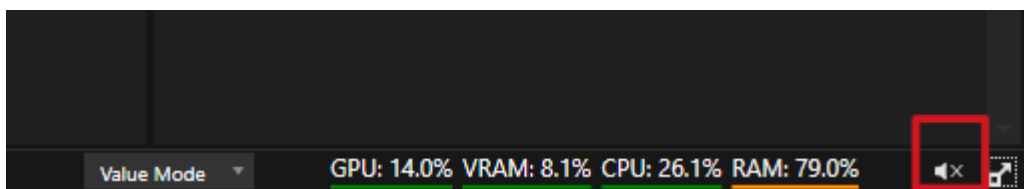
#### **Using Dante PCI or USB Audio Devices:**

Use Yamaha products only with Intel CPUs as AMD CPUs are not supported. Yamaha products have per default 128 channels activated. You might need to change this in the project settings in order to avoid performance issues.

When using Dante Virtual Soundcard make sure you set it to 16bit and 48kHz.

Recommended settings for ASIO devices is 16bit/ 48kHz with a buffer size of 1024 or higher. Focusrite USB devices may require ASIO lock buffers.

## Mute Preview Audio for local System



The **status bar** provides you quick access to **mute preview audio** on a VERTEX system

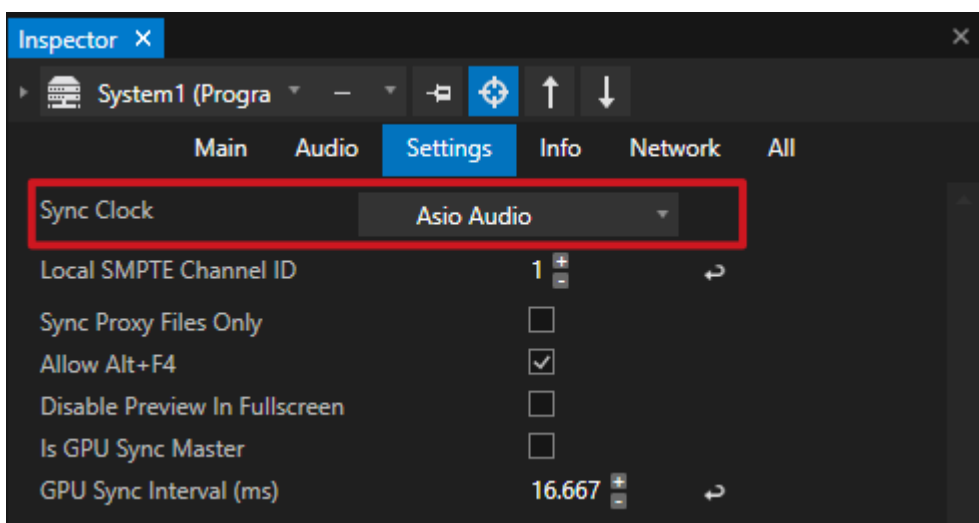
## Channel Routing

Audio channel routing **is done on clip container or track level.**

Audio from each single clip container can be routed freely to the available channels of the audio device.

Please read topic [Audio in the chapter Working With Content](#).

## Sync Clock Settings

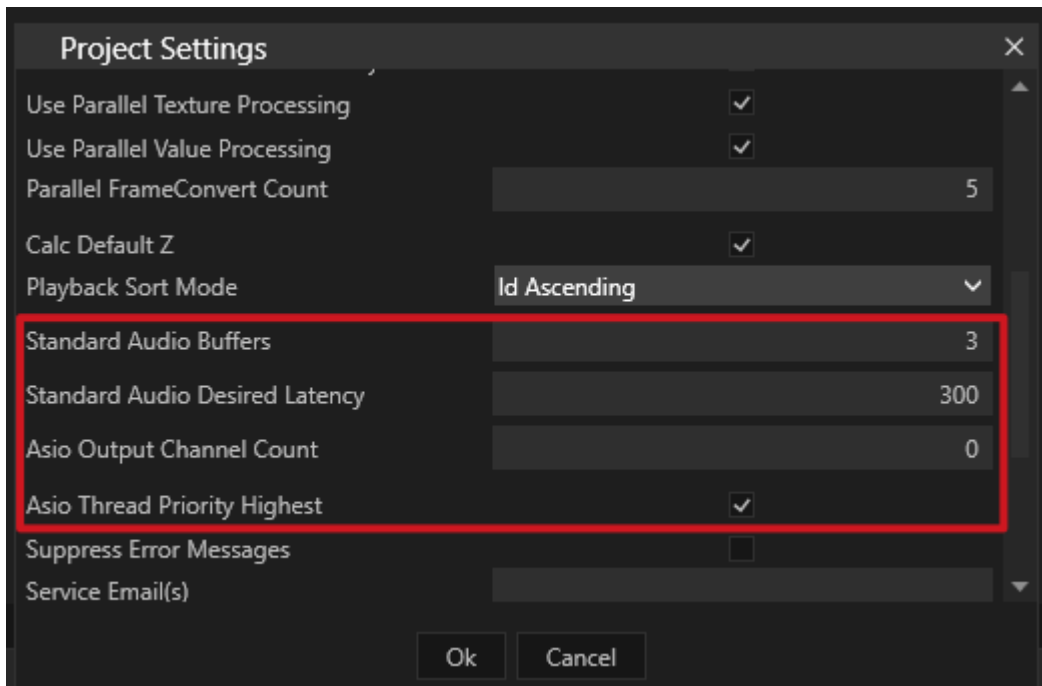


When switching between audio devices, VERTEX also changes the sync clock source.

Assume that the audio output generates the clock for a playback.

Please double check the settings and the clock source.

## Advanced Audio Settings



Open **project settings** to set advanced audio parameters.

#### **Buffer and Latency for Standard Audio**

Set to optimize the audio playback quality for your standard audio hardware.

#### **ASIO Output Channel Count**

Default: 0 - channel number is taken from ASIO device driver.

Set to a value if number of your channels was not detected right by VERTEX or to reduce channels due to performance reasons ( e.g. for DANTE physical or virtual sound cards).

Set back to 0 to reset.

#### **ASIO Thread Priority Highest**

Enabled per default: ASIO processing gets highest Windows 10 thread priority to ensure best audio processing and playback performance.

Disable for a better system performance. ASIO audio processing will not be done with highest thread priority anymore.

## 5.3 Content Editing, Composition & Playout

- Choose whether you want to arrange your content in a **timeline based [sequence](#)** or with a **clip based [playlist](#)**.
- Create multiple **playbacks** that **can either host a sequence or a playlist**
- The **[Playback Mixing Engine](#)** or **PME** is VERTEX's video mixer.

### The Composition Of Content On Canvas

- Whether you like to arrange the order of your content in a playlist or or a timeline based sequence, the positioning of all content happens on a Canvas.
- Feel free to look up the introduction to the basics of [Canvas, Surface and Output](#)
- or read more about the details in the following chapters on [Canvas](#) and [Surface](#).

## Sequence, Playlist, Playback and PME



## Sequence Or Playlist As Playback Provider

Sequences and a playlists are so-called "playback providers" that have a defined length and are hosted by a Playback.

### Learn more about Sequences

- [Sequence](#)
- [Clip Container](#)
- [Keyframes](#)

### Learn more about Playlists

- [Playlist](#)
- [Clip](#)

## Playback

A Playback is the host for your playback provider - a sequence or a playlist.

The playback handles the transport functions PLAY, PAUSE and STOP. It defines the mode how your playlist or sequence is played (Loop, Once, Reverse)

and is responsible for timecode handling. Each Playback has a mixing level to fade in or out.

It is possible to change the playback provider anytime from sequence to playlist or another sequence.



When a new Sequence or a Playlist is created, VERTEX automatically creates a new Playback.

### Learn more about Playback settings:

- [Playback](#)

## Playback Mixing Engine (PME)

PMEs host all Playbacks of a project. - In terms of a video mixer, PME's are your master groups or faders.

By default there are 2 PME's: **Live** and **Preview**.

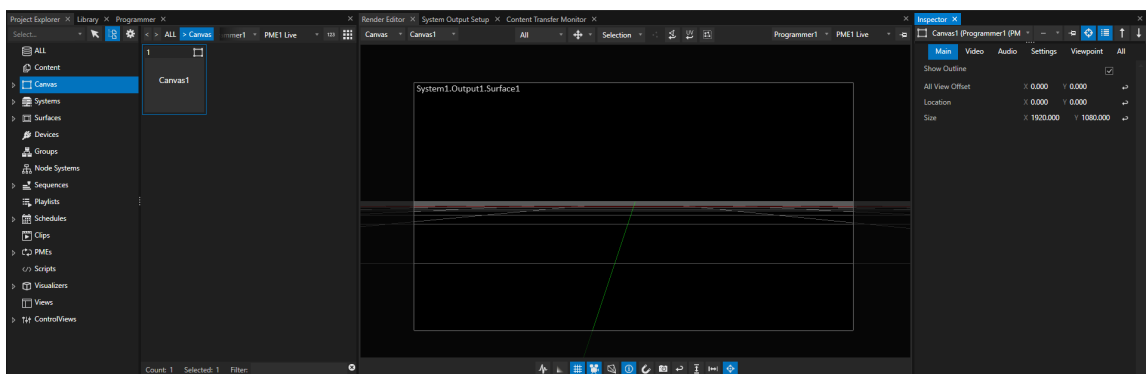
Route or fade any of your playbacks either to a **live output** for your audience, **or to a preview output** for an operator.

Learn more about [PMEs here](#).

## 5.3.1 Canvas

- A Canvas **defines your working area in VERTEX** - like a canvas **for your painting**.
- It is possible to have **multiple Canvases** in one VERTEX project - for instance one for each video wall of your stage, or one for each room.
- **The target Canvas** of each **Clip Container** ( or a playlist's **Clip**) can be changed on the fly. This way, it is easy to re-arrange content in your [Canvas-Surface-Output](#) setup.

## Canvas Model in VERTEX



- Every new VERTEX project **starts with one Canvas** in the size of your local Windows desktop.
  - The **Canvas size** is adjustable anytime in the Inspector
  - and so is the **number of Canvases**.
  - By default, the **coordinates zero position is the middle of the Canvas**. Per Canvas or globally in project settings, you are able to **set this point to the top left corner** of a Canvas
- 
- **Add a Surface** to a Canvas per drag-and-drop from Project Explorer or via context menu from the Render Editor.
  - Please also read [about the Canvas-Surface relation](#).



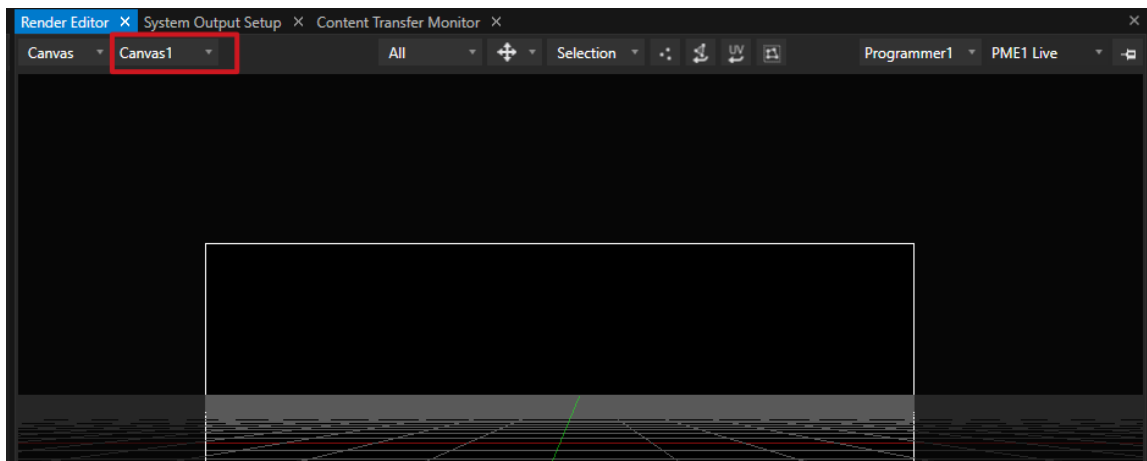
### Canvas Space



### Output Space



## Select and show Canvas in Render Editor



- Go to "Canvas View" in the Render Editor
- Use the drop-down menu to select the Canvas that needs to be shown.



When having a big Canvas with multiple Surface from different Systems, please keep in mind:  
In the current version of VERTEX all Clip containers that are assigned to this Canvas are rendered on all assigned Systems. For splitting the render performance, we recommend to use a Canvas for each System.

## 5.3.2 Playback

- A **Playback is the host** for your **Playback Provider**: a **Sequence** or a **Playlist**
- The Playback handles **play, pause and stop**. It defines the **mode** how your Playlist or Sequence is played (Loop, Once, Reverse) and is responsible for **timecode** handling.
- The **Playback Editor** is the place where you can arrange Content into a Playback Provider
- **Depending on the Playback Provider, the UI of the Playback Editor changes: a Timeline Interface** for Sequence or a **Playlist Interfaces** for a Playlist

## Overview: Playbacks in VERTEX



A **Playback** is the host for a **Playback Provider** - this could be:

- a [Sequence](#)

or

- a [Playlist](#)



Speed up the daily work: VERTEX automatically creates a **Playback** for every **Sequence** or **Playlist** you create.  
The base settings are done for you..

The **Playback** handles **play** , **pause** and **stop**. It defines the **mode** how your **Playlist** or **Sequence** is **played** (e.g. **Loop**, **Once**, **Reverse**)

and is **responsible for timecode handling**. Each **Playback** has a **mixing level** to fade in or out.

You are able to **switch a Playback Provider** of a **Playback** and change it from **Sequence** to e.g. a **Playlist** or another **Sequence**.



**The length of a Playback is defined by the Playback Provider.**

By default the length is set as 10 Minutes.

If you want to change the length of your timeline for a sequence or the displayed frame scale - you have to set this for your Playback Provider. Select the hosted Sequence into Inspector

The **arrangement of the content** itself is done into the Playback Provider (Sequence or Playlist)  
Each Playback is available in every **Playback Mixing Engine (PME)**. PME acting as "Video Master Faders".  
They control the main playout level for the whole mix. By default for live and for one preview.

The **Playback Editor** is the place to edit and compose your content. Depending on the Playback provider you has set, the **UI of the playback editor changes**. You will get a

- Timeline Interface for a Sequence
- Playlist Interface for a Playlist

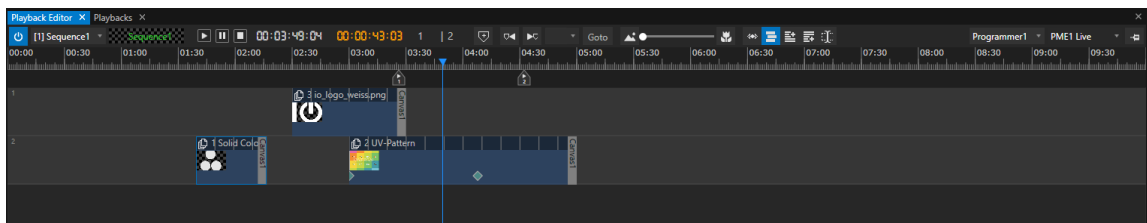
## Working with Playbacks

To edit, overview and control Playbacks there are 3 main editors and points into VERTEX.  
All Settings for a Playback are done as usual into the Inspector.

### Playback Editor

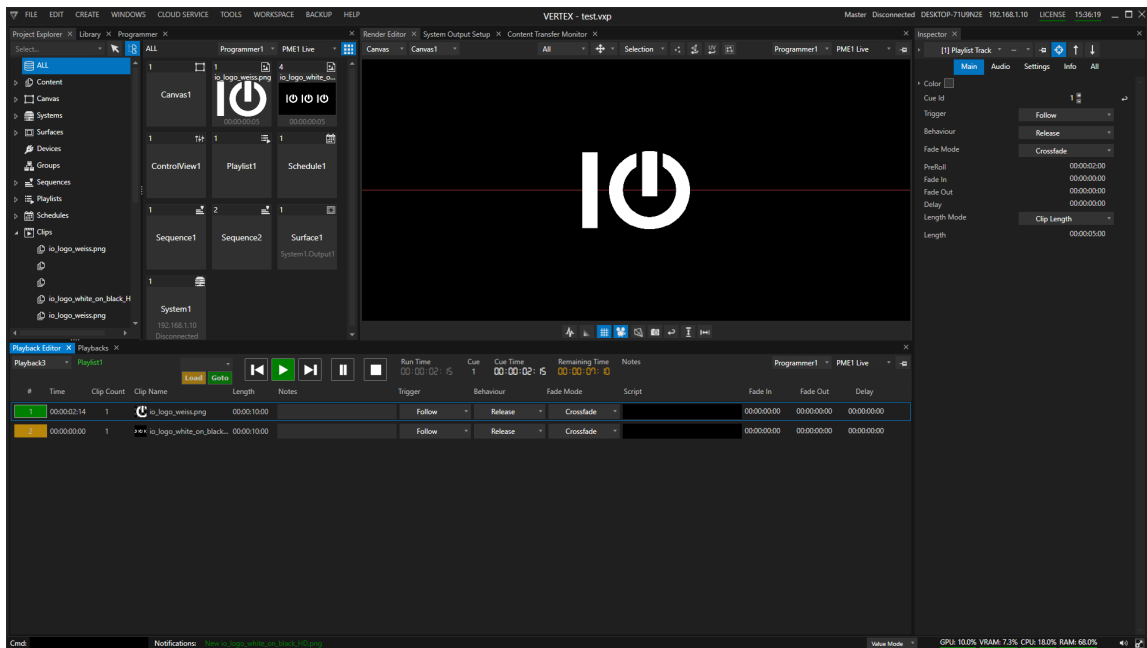
The Playback Editor is the main tab where you can arrange and edit the Content of your playback provider  
Learn more basics about the [Playback Editors User Interface](#)

**Playback Editor UI for a Sequence:**



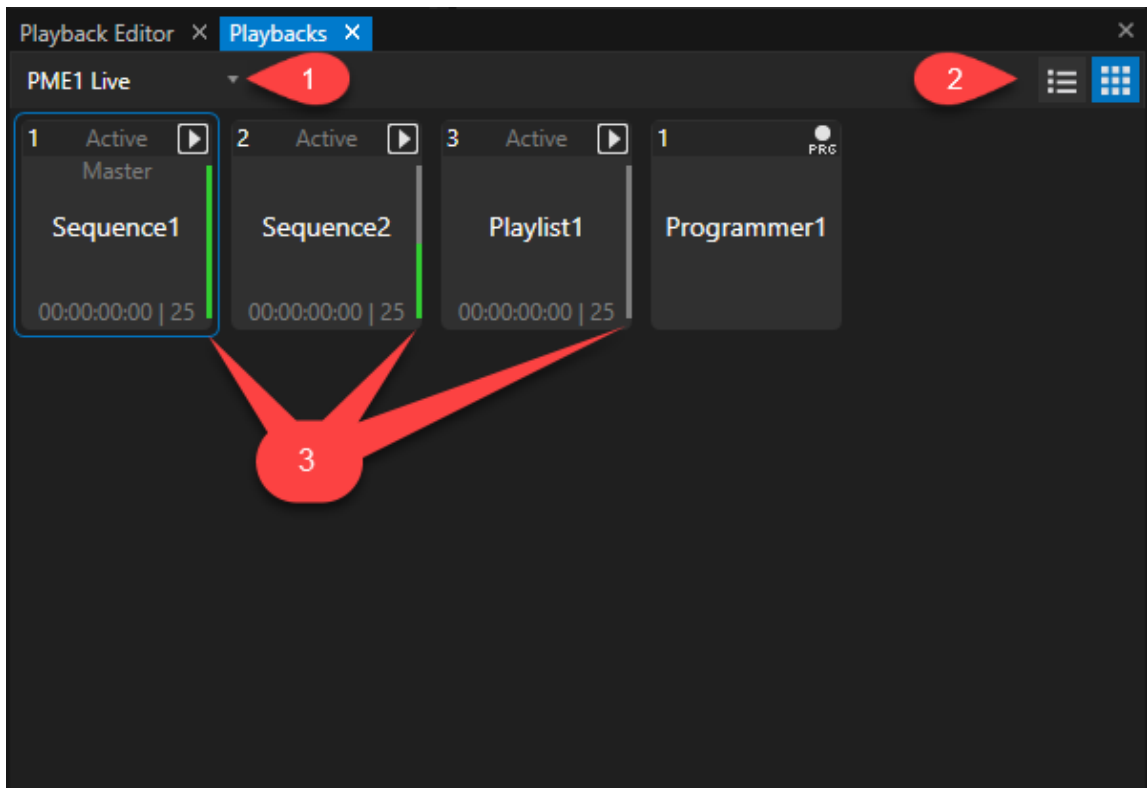
Please go to the [Sequence Topic](#) to learn details about the timeline based workflow with Clip Containers

**Playback Editor UI for a Playlist**



Learn more about [handling and options of a Playlist](#)

## Playbacks Window



The [Playbacks-Window](#) gives you a quick overview about

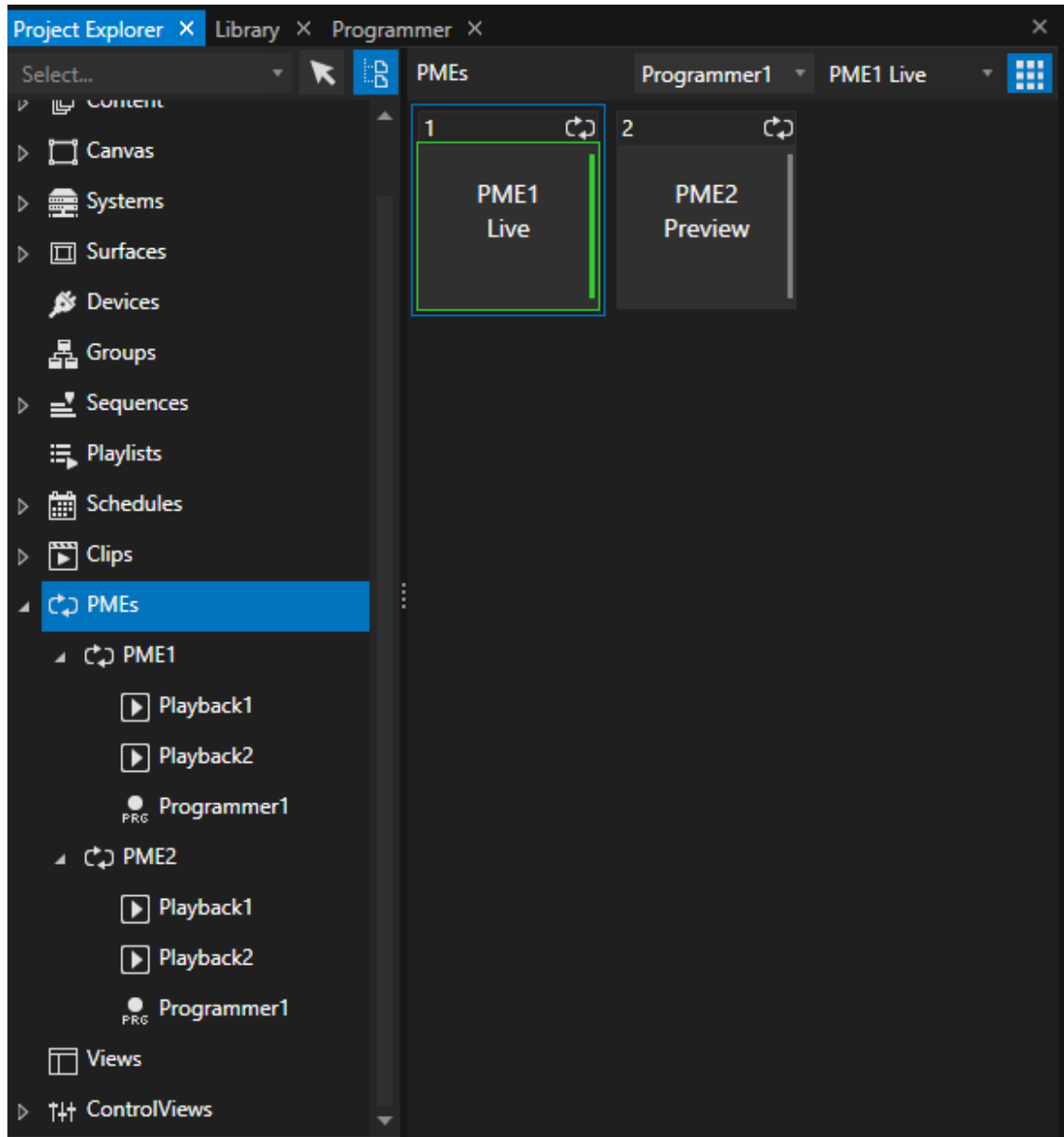
- all existing Playbacks into your Project
- their Playback Provider
- their current status
- their mixing level.

With a rightclick you are able to open a context menu to set Playback Mode and Speed  
Selecting a Playback will focus it into Inspector

## Project Explorer

Last but not least, the Project Explorer always is a good point to show and select Playbacks for settings and to get a quick overview about their state and their mixing levels

Playbacks are located as childs of a Playback Mixing Engine (PME) there



## Assign a Playback provider to a Playback

When you create a new Sequence or a new Playlist, VERTEX creates also a new Playback for you and directly assigns Your Playlist or Sequence to this Playback.

**Use Case:**

*Playback 2 hosts a Sequence. During your Project work it seems like a Playlist would be a better solution to arrange your content.*

*Best for you would be to use Playback 2, because you have already created some Control Buttons in a Control View.*

*You want to change the Playback Provider of Playback 2 from Sequence 2 to Playlist 1.*

*There are three points to change or assign a Playback Provider to a Playback:*

## Inspector

*Focus or Pin your Playback into Inspector*

*Drag with your mouse a Sequence or a Playlist from Playback Editor to the Playback provider Field into Inspector*

## Playback Editor

*Select your Playback into Playback Editor*

*Drag with your Mouse a Playlist or a Sequence in the Playback Provider info there.*

## Playbacks Window

*Open a Playbacks Window or navigate to the already opened Playbacks tab*

*Drag a Sequence or a Playlist from Project Explorer to a Playback*

## Playback order into Fullscreen or Render Editor

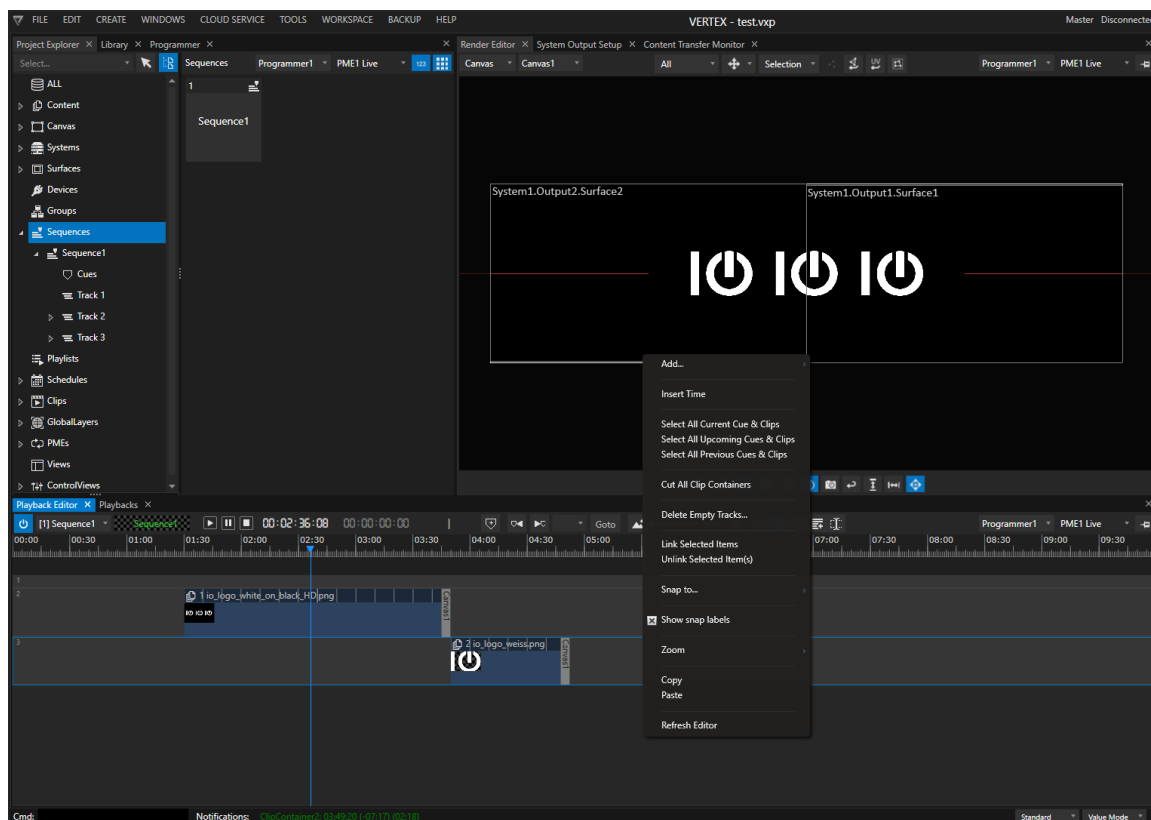
- *Playback Order: By default ascending.*  
*When Playback 1 and 2 are running in parallel and are assigned to the same Canvas:*  
*Means Playback 2 is in front of playback 1*
- *you are able to change this order into project settings*  
*Options: ascending, descending or last change (LTP)*



### 5.3.3 Sequence

- A Sequence is a **timeline based Playback Provider** that is **hosted by a Playback**
- A Sequence **contains a collection of clip containers (and cues), sorted on different tracks on a timeline**
- A Sequence has a **default Canvas**, but every clip container could be assigned to a different Canvas or even output
- A Sequence **could be nested as clip container** into another sequence - similar to what you know from subcompositions in other Compositing Software

## Working with Sequences



### Create a new Sequence

- via the main menu on top: Open "Create" and select "Sequence"

- *via the Project Explorer: navigate to the Sequence Category, open context- menu with a right-click and select "Create..."*

## Clip Containers on Tracks

*A Sequence contains different Clip Containers. Tracks help you to arrange Clip Containers and build your show. Freely move Clip Containers between tracks.*



### **Too much tracks?**

*Scroll up and down vertical with your mouse wheel. Use + and - to zoom in or out into Playback Editor*



### **Are tracks the same like layers which I already know from other media server software?**

*No! Tracks are just helpers to order your Clip Containers. Tracks can have some track based settings but there is no layer restriction like other software has. Tracks and Clip Containers are flexible objects - like you may already know from Video editing or compositing software*

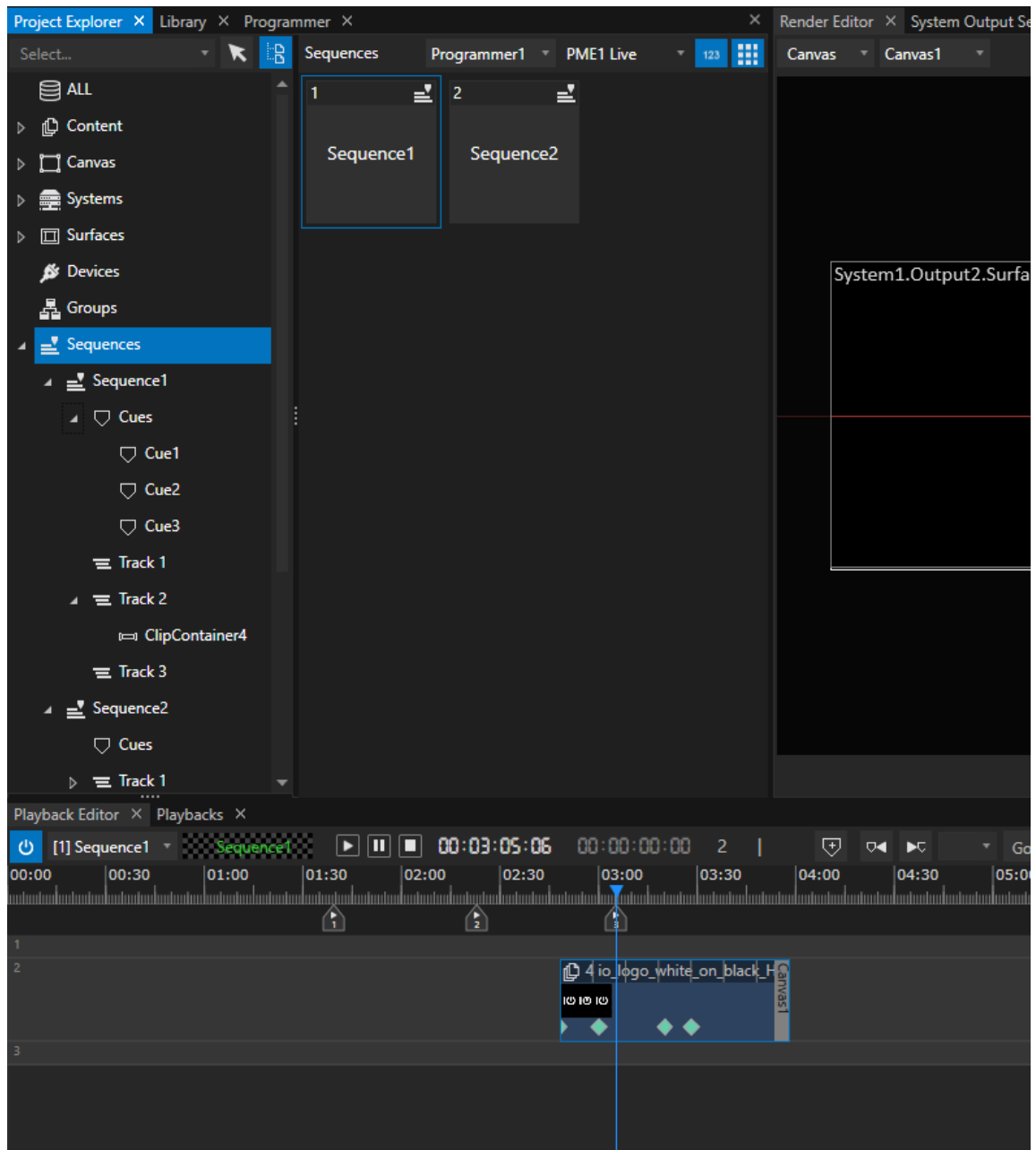
## Change Length and Framerate

- *By default, the length of each Sequence is set to 10 Minutes.*
- *You are able to change this length into Inspector Settings of your Sequence:  
Select the Sequence into Project Explorer.  
Focus it into an Inspector and set the Playback provider time into the settings tab there*

## Options in the Context menu

*By a right-click with your mouse into the Playback editor, the context menu offers you different options to select clip Containers, create or insert new tracks or to create other items.*

## Access all items of a Sequence into Project Explorer



All tracks, Cues and clip containers are also accessible as child elements of your Sequence into Project Explorer. You are able to select them there to focus an item into Inspector or to delete.



### **Lost Clip Containers**

*Searching for a Clip Container that is lost? Or too small to select into Playback Editor?*

*Just select it into Project Explorer and open the context menu.*

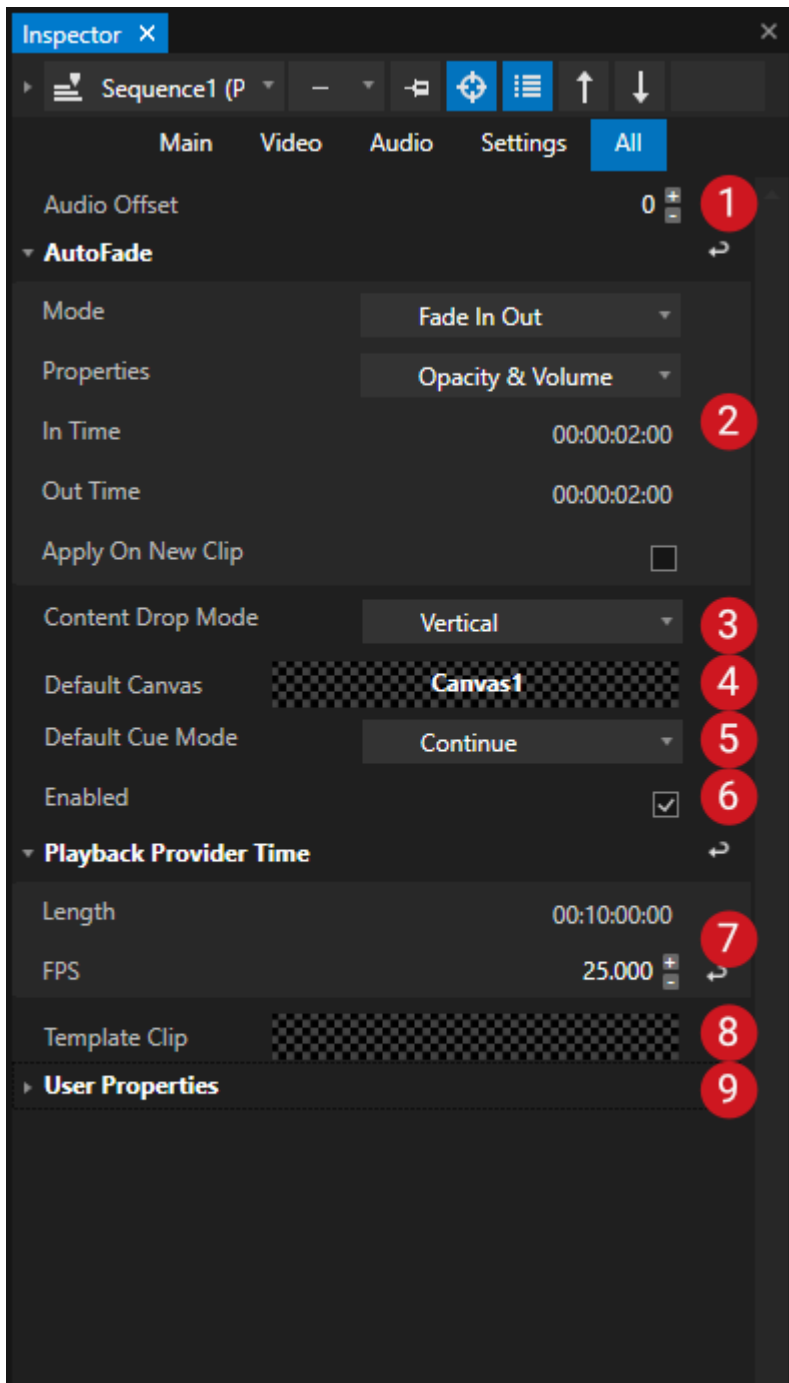
*The option "reset to clip length" resets the Container to its default value again.*

## Cues

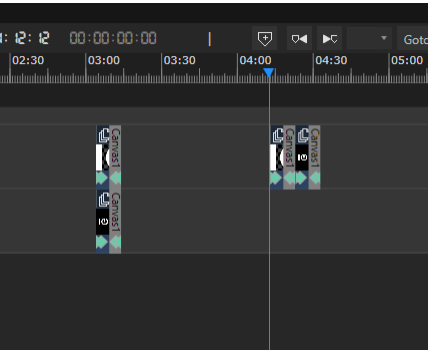
Read more about Cues and Cue handling [here](#)

## Settings

## Standard Mode



<b>1</b>	<b>Audio Offset</b>	<p><i>Defines a Audio Offset for your whole Sequence in milliseconds. All Audio is shifted from video for this offset</i></p> <p><i>Set to a value to compensate a general audio shift from video content</i></p> <p><i>Please note: There is also an option to set this offset per System</i></p>
----------	---------------------	--

2	<b>AutoFade</b>	<p><i>Auto Fade Options and Settings.</i></p> <p><i>For each Clip Container into a Sequence there is an Auto Fade Wizard which automatically generates Keyframes to fade in/out video/audio. Set default settings for the whole Sequence here. Enable "Apply on new Clip" to add this Keyframes for each new Clip Container</i></p>
3	<b>Content Drop Mode</b>	<p><i>Defines how defines how a selection of content that is dragged from Project Explorer into your Sequence will be arranged as Clip Containers.</i></p> <p><b>Options:</b></p> <p><i>Vertical (Default): As Clip Containers, each on a new track at the same timecode</i></p> <p><i>Horizontal: ext to each other on the same Track</i></p> 
4	<b>Default Canvas</b>	<p><i>Specifies the <u>Canvas</u> that is set as default for each new <u>Clip Container</u> into this Sequence.</i></p> <p><i>You can change this Canvas by drag and drop another Canvas with your Mouse from Project Explorer to this Property field.</i></p>
5	<b>Default Cue Mode</b>	<p><i>Sets a default <u>Cue Mode</u> for this Sequence. Every new Cue which will be added to this Sequence will have this default mode</i></p>
6	<b>Enabled</b>	<p><i>Enables or Disables rendering of the whole Sequence</i></p>
7	<b><u>Playback Provider Time</u></b>	<p><b>Length:</b> <i>Defines the length of your Sequence. After a change of this value also the timeline length into Playback editor is adapted to this new value</i></p> <p><b>FPS:</b> <i>Defines the Frames per Second for the Playback Editor's timeline. This value only affects the User Interface and its fine scale-. The Renderer itself pushes all frames of your content to your GPU - depending of your contents FPS and the frequency that is set of your display</i></p>

8	<a href="#">Template Clip</a>	Set a <a href="#">Template</a> for a Clip Container. For each new Content that will be added to the Sequence a new Clip Container based on this template will be created. Drag a Clip with your mouse from Project Explorer to this property field.
9	<a href="#">User Properties</a>	Enter some <a href="#">User Properties</a> for this Sequence



### **Changing the sequence framerate may shift already existing cues!**

When changing the frame rate of a sequence that already has cues in the timeline, all cues will be shifted to the next valid time code frame.

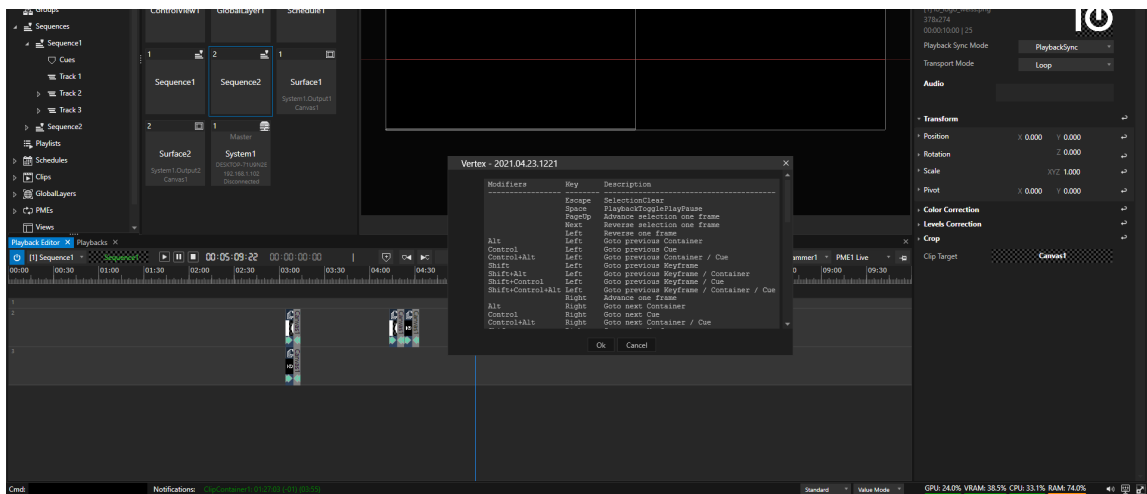
Depending on your changes, this recalculation may cause a minimal shift in cues - please inspect all your cue times after changing the framerate.

## Advanced

Change [Inspector Mode to Advanced](#) to display all advanced settings for a Sequence

## Shortcuts

- There are **seperate lists of all available shortcuts** for each editor and window in VERTEX.
- To keep the documentation up to date, the **shortcut lists are automatically created by the software itself**.
- Just **press the F1 Key** to open a list of shortcuts corresponding to the current window/ editor in focus.



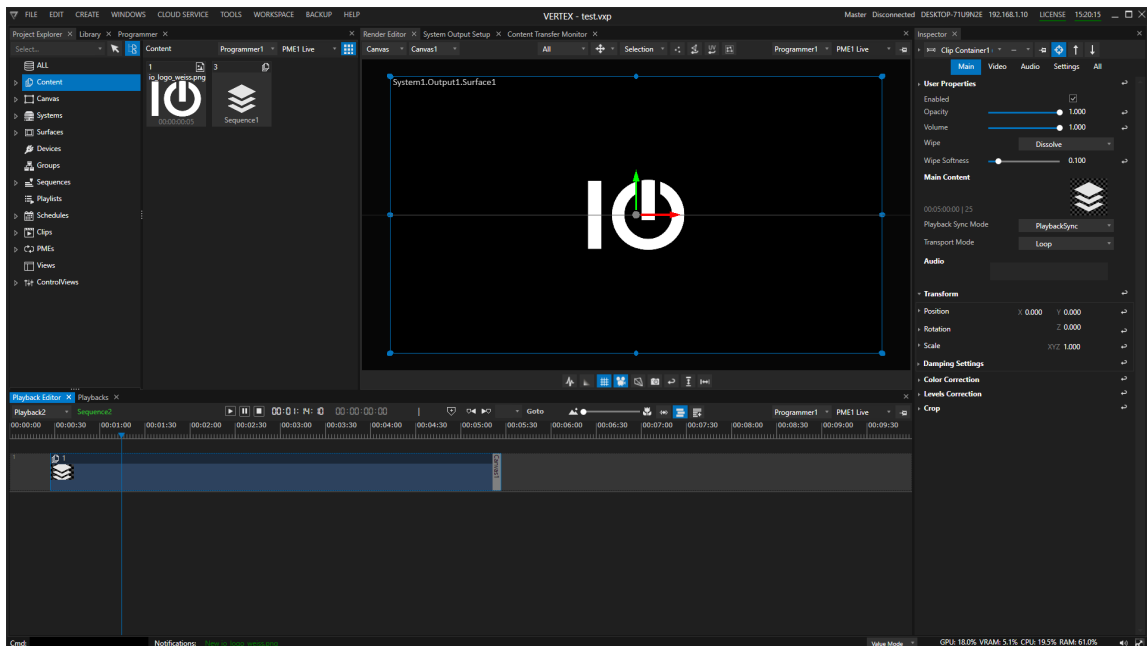
*Focus Playback Editor and Press F1 Key.  
A window with all available shortcuts for your Sequence opens.*

## Nested Sequence Content

- VERTEX allows you to use **a Sequence as Content into another Sequence**.
- This gives you as user the capability to make **sub-compositions** and use them as content into another sequence
- The capability and the performance limit of nesting depend on your hardware. In general: **Nested sequences need more render performance** as regular simple video content.

## How To Create





There are **two different ways** to nest a Sequence into another one:

**1. Drag a Sequence with your mouse from Project Explorer into the Playback Editor.**

VERTEX creates a Clip Container for this Sequence and a Nested Sequence Content into the content section of your Project Explorer.

You can identify a Nested Sequence Content by the layer symbol



**Restrictions for Nesting**

Currently and all kind of Devices are **not supported** for Nested Sequence Content.

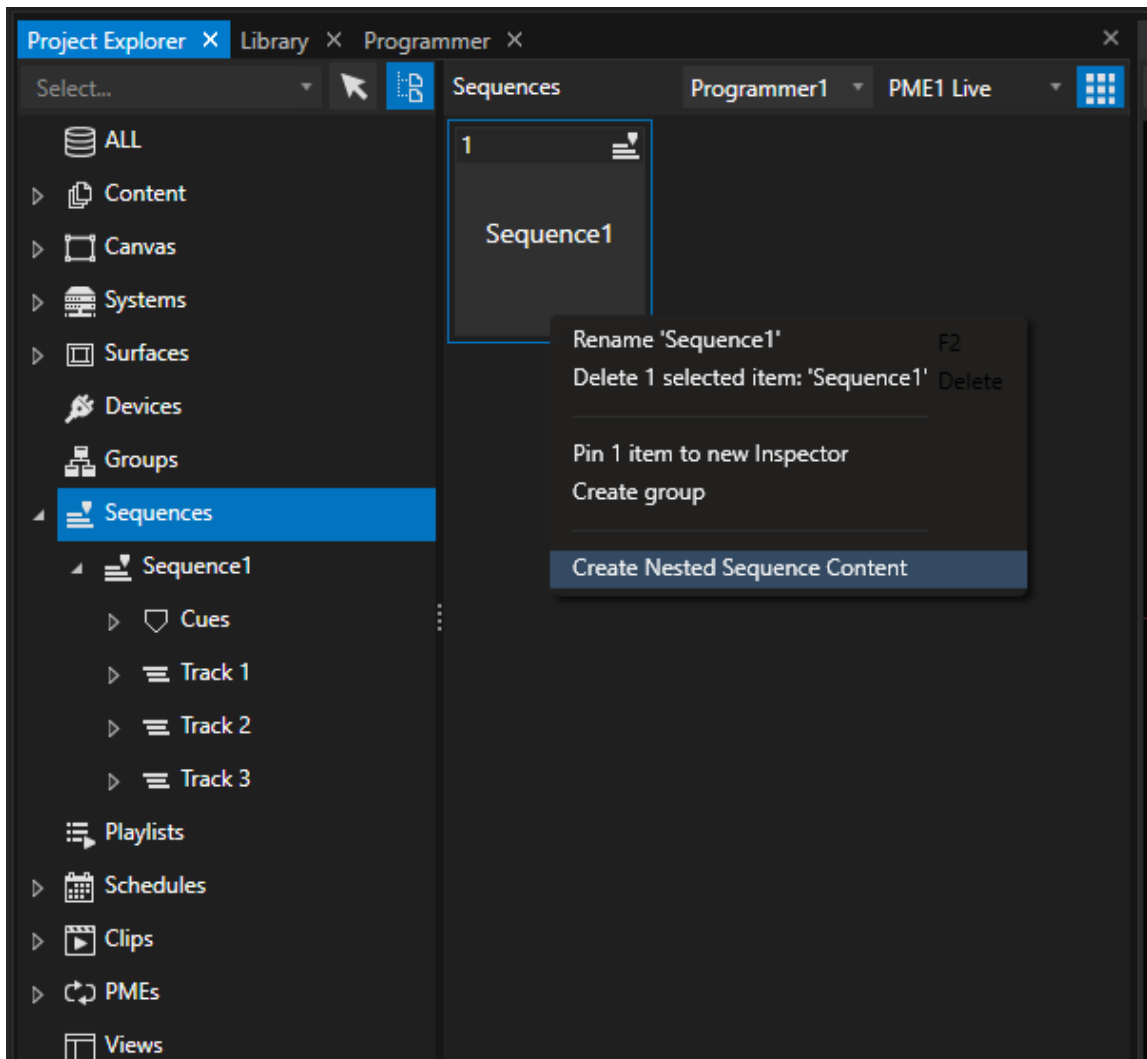
We will work on this for future release.

A **circular reference is not possible**: Nested Sequence Content from a Sequence 1 cannot be used into Sequence 1.

**2. Use the Context menu into Project Explorer and create a Nested Sequence Content**

One created, this Nested Sequence Content is listed into the Content Section of your Project Explorer and can be used

as every other Content into a Clip Container of another Sequence



#### **Performance**

*VERTEX is a real time software - please keep in mind that using nested sequence content costs you more System performance.*

*Each nested Sequence has impact on the Render Pipeline. The performance needs of the Render Engine increases with deeper nesting, repeated or multiple nesting.*

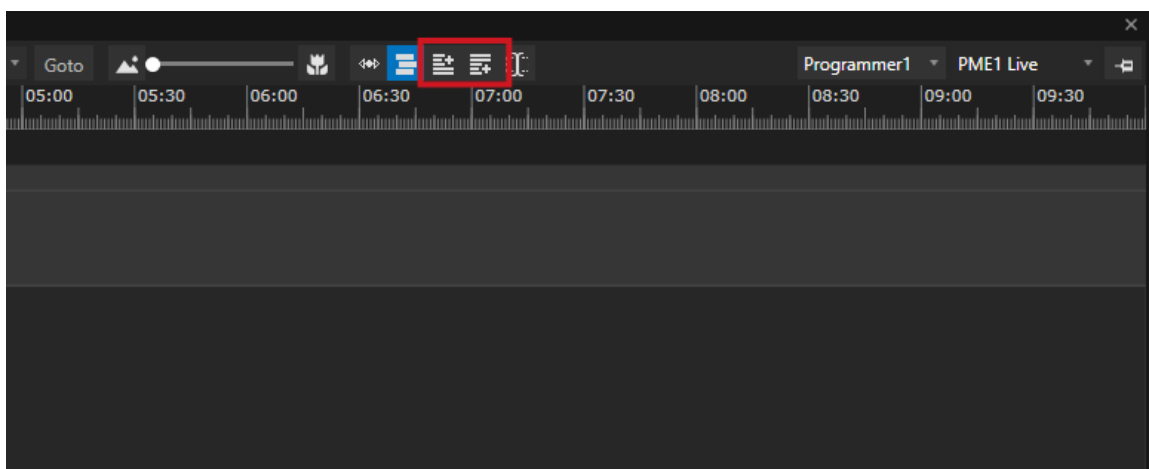
### 5.3.4 Track

- A Track is a **visual helper to structure and order** Clip Containers
- You are able to **freely move Clip Containers between tracks**
- A Track **can host multiple Clip containers**

## Add Track

There are 2 ways to add a track:

*Add Track Icon into Playback Editor UI*



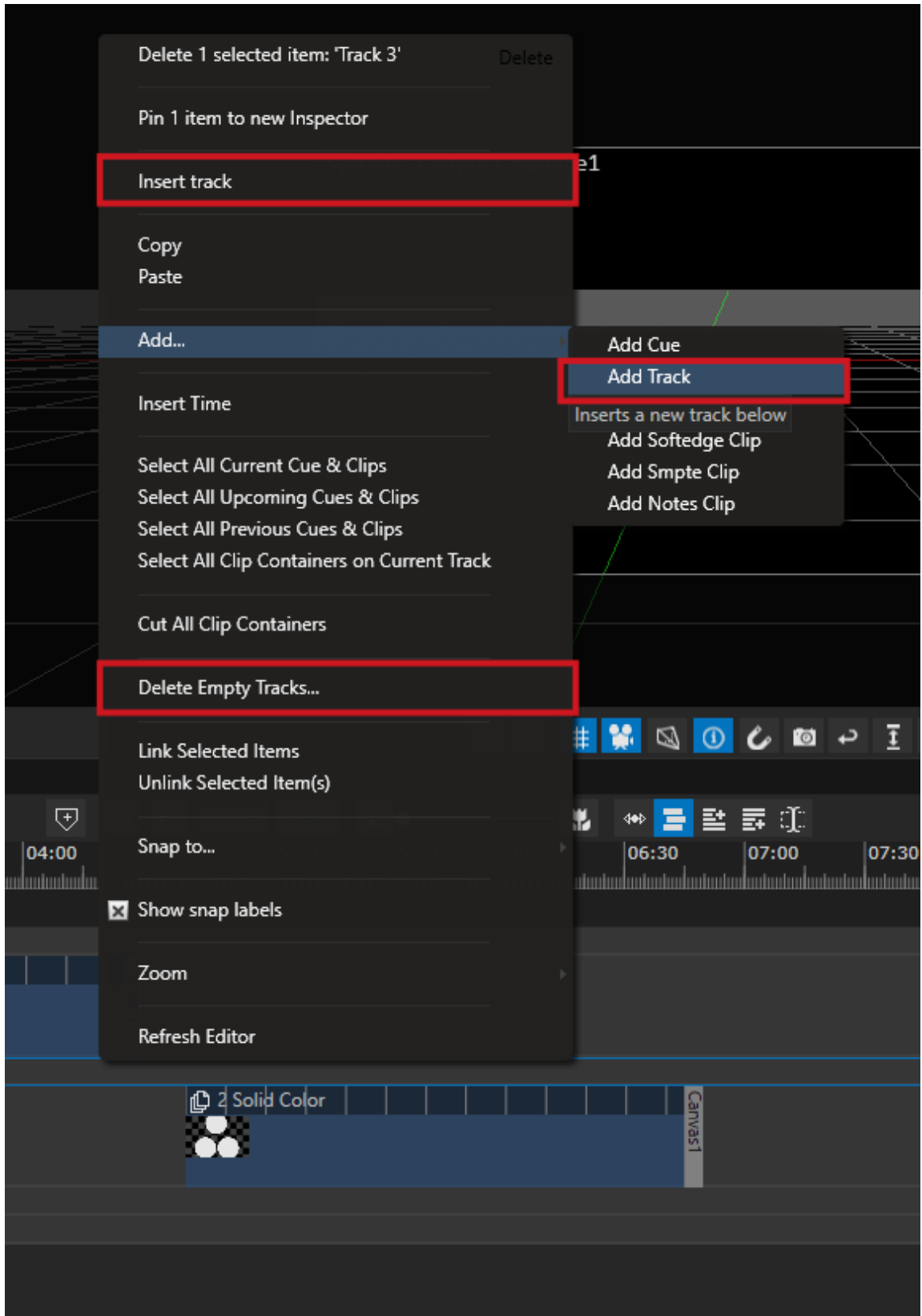
- The first of the two +-buttons adds a new track on the top
- The second + button adds a new track at the bottom below all other tracks



***Are tracks the same like layers which I already know from other media server software?***

*No! Tracks are just helpers to order your Clip Containers. Tracks can have some track based settings but there is no layer restriction like other software has. Tracks and Clip Containers are flexible objects - like you may already know from Video editing or compositing software*

## Context Menu into Playback Editor



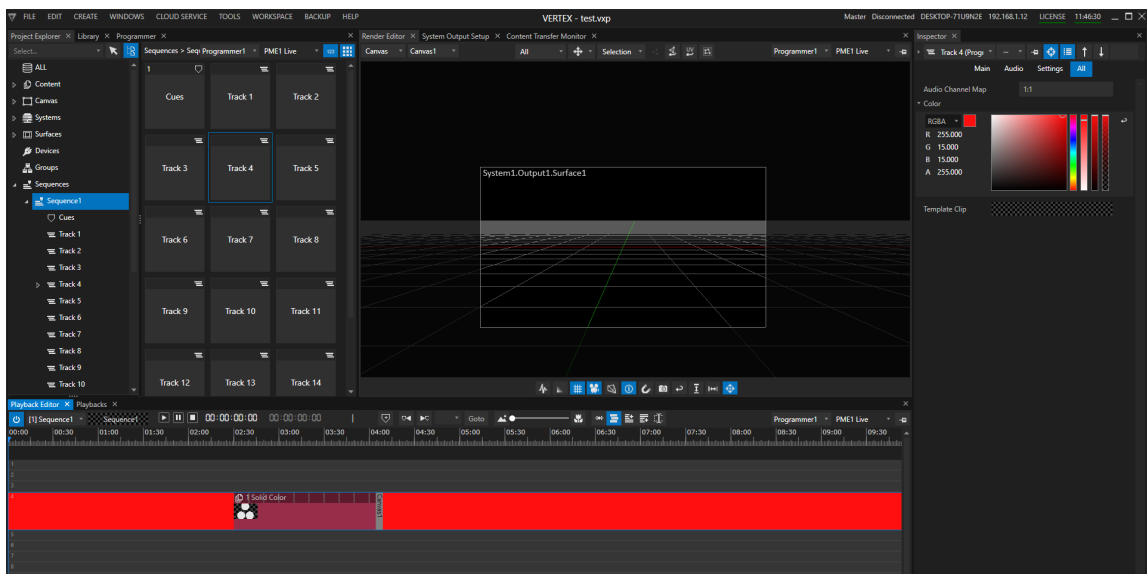
- Open the **Context Menu** with a right-click into Playback Editor
- There are **two options** to add a new Track:
  - Navigate to "add" and select **"Add Track"** - a new track is added at the bottom below all other tracks
  - If there already is another Track, you will see the option **"Insert Track"**. Use this option to add a track above the one you have focused when you open the context menu
- Use **"Delete Empty Tracks"** to delete all empty tracks of your current Sequence



### Scrolling and zoom

Scroll up and down with your mouse wheel to navigate through an huge amount of tracks  
Use + and - to zoom in or out into Playback Editor

## Settings



### How to select a Track into Inspector

There are two options how to select a track into the Inspector

#### Option 1

- Left-Click with your mouse on a track into the Playback Editor (not on a Clip Container on this track!)

#### Option 2

- Go to Project Explorer
- Select your current Sequence
- Open child elements into Project Tree
- Select a Track there
- Now your Track is focused into Inspector

## Color

Sets a user specific Track Color

## Audio Channel Map

With this setting you are able to route Audio Channels per Track.

This setting is valid for all Clip Containers on this Track.

Enter a Routing by using the Syntax below:

- 1:1 - Default Value, no routing
- 1@4,2@7 - Audio Channel 1 is routed to Audio Output Channel 4 of the selected Live Audio Interface, Channel 2 is routed to Output Channel 7

## Template Clip

Assign a Clip Template that is taken into account as blueprint for every new Clip Container on this track.

Read more about [Clip Templates here](#)

### 5.3.5 Clip Container

- **Clip Container** can host content, settings, device controller data or even work as a notepad.
- Clip containers are **free from layer restrictions** and **can be moved around freely in the timeline**.
- Clip containers are arranged **on tracks** in the timeline.

## THE VARIOUS KINDS OF CLIP CONTAINER

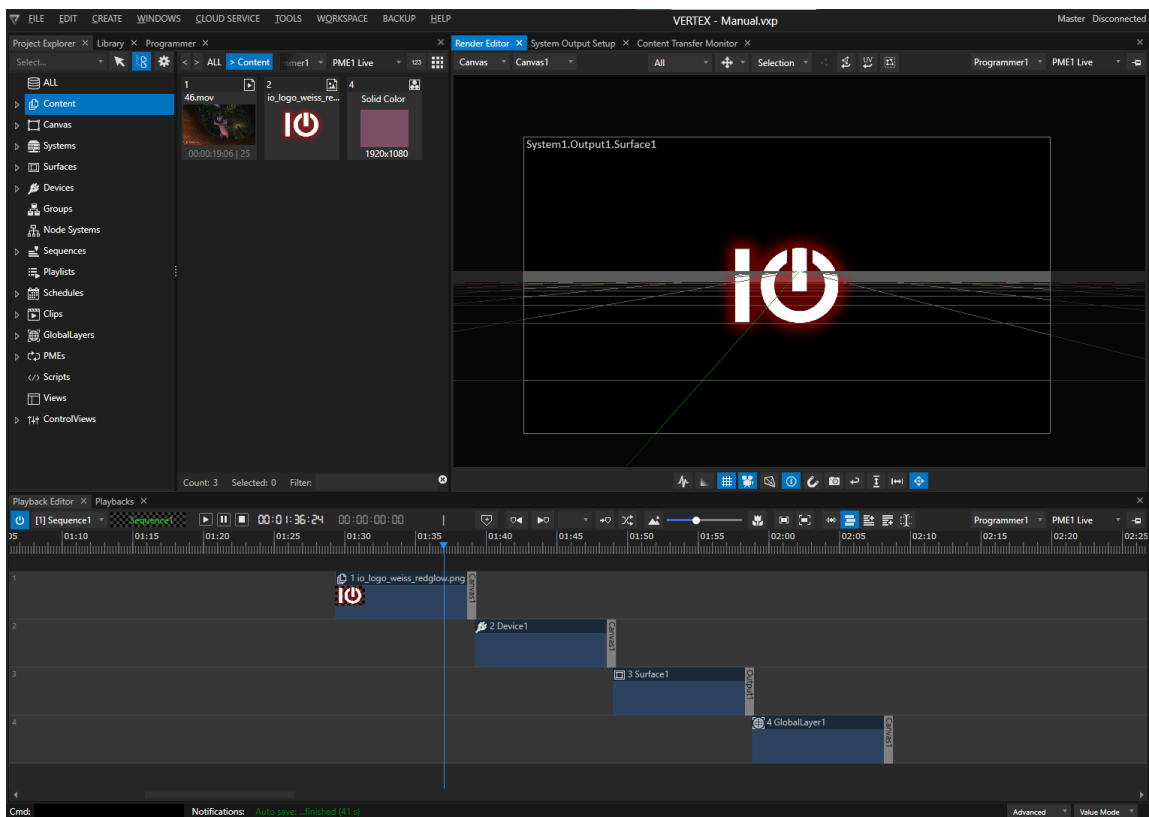
There are different types of clip containers available - most of the settings are similar for all - some of them have special parameters and properties:

1. **Clip containers for content** ( most flexible type that you will use frequently for almost everything)
2. And **other types of clip containers** such as

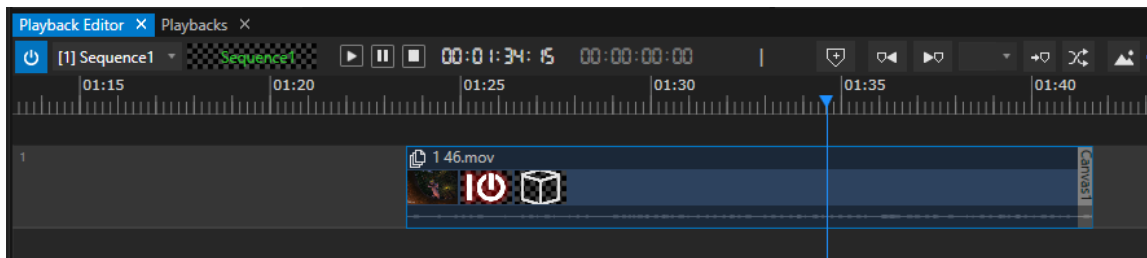


- o clip containers that control **devices** (i.e. DMX devices)
- o clip containers that **overwrite global settings for surfaces and outputs**
- o clip containers for **special elements like global-layers or console layers**
- o clip container just for **notes**
- o **SMPTE clips** that generate timecode

Some of them (like those for content) you will use more often than others. Some of them are useful for edge cases .



## CLIP CONTAINER TARGET



Clip Container with Canvas 1 as Target

## Every clip container has got a target:

- This target is displayed **as a flag at the end of each clip container**
- For content clips this **target is usually a canvas**. In special cases, this **target could also be an output or even a surface**. Or just another canvas.
- Different clip containers in the same sequence **can have different targets**- there is no restriction. For instance, when working with 3 different canvases, you are able to set them as different targets for your clip containers in the same sequence.

## Add or change a target

1. **Drag** a canvas, a surface or an output with your mouse **from the project explorer** and **drop it into a clip container**.
2. Select a clip container with your mouse, go to settings tab in the **inspector**, drag another target from the project explorer and drop it into this property field.



### **Change The Default Canvas Of A Sequence**

Select a sequence in the inspector and change the default canvas in the settings tab there. Every new clip container in this sequence will have this new canvas as target.

# CLIP CONTAINERS FOR CONTENT

## Content types

A content clip container can host one of the content assets below:

- audio
- image
- an image sequence

- video
- generative/procedural content such as text, solid colors, gradients or test patterns
- shared textures
- nested sequence content
- HTML browser content

## Additions

In addition to one of the content types above you are able to assign **from the project explorer**:

- a mask
- a separated audio track
- a 3D object

or **from the project library**

- Video Effects (FX)

## Workflow and Settings

- **Create a clip container**: just drag and drop content from the project explorer into the playback editor.
- **Replace content** by dragging another content file from the project explorer into the clip container
- Add **video FX**: drag and drop your desired effects from the library into the clip container



**FX always requires a base texture in the clip container to work.**

This texture can be every type of content - To run an effect like a particle shader use a solid color content as texture for the clip container.

- Add audio to a clip container with video or image content inside: Just drag the audio file from the project explorer into the clip container.  
This action also will **replace the embedded audio file** from a video in the clip container with the new one.
- **Add a mask** to a clip container with already video or images inside: Hold Shift key and drag the mask image to the clip container.
- Drag a **3D object** with your mouse from project explorer to a clip container to add.

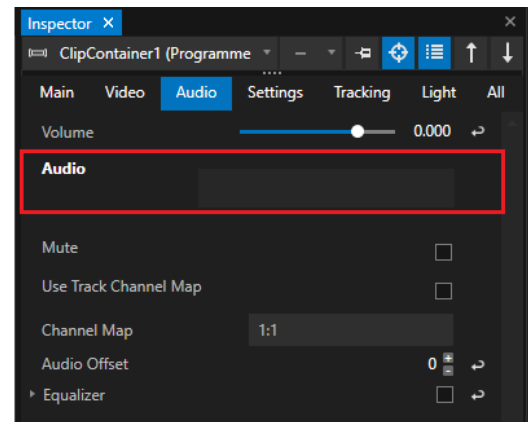
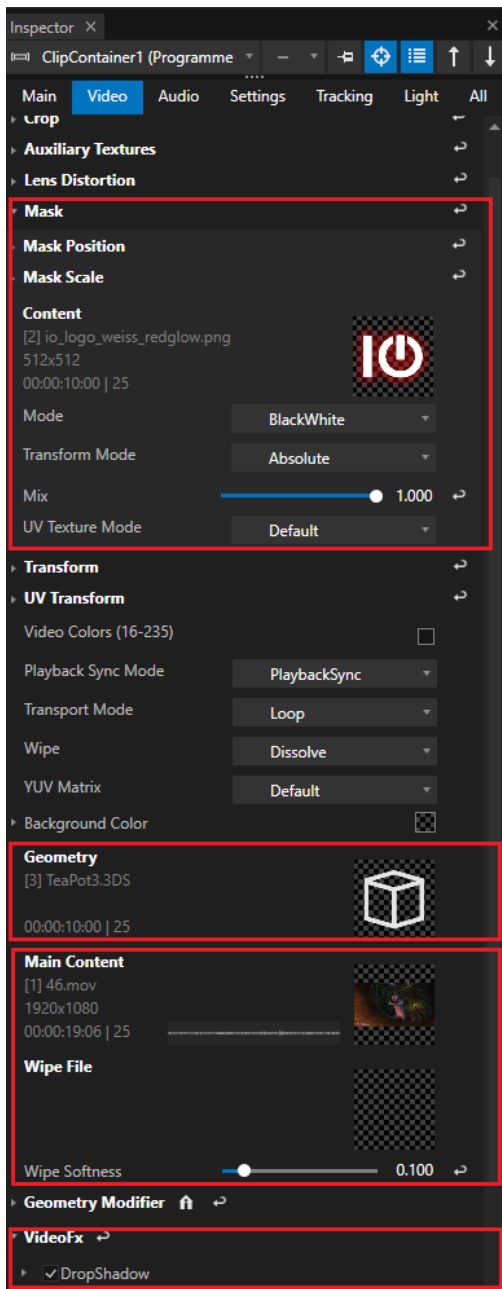
**3D Objects always require a texture**

*3D Objects in a Clip Container always require a texture. This texture could be every type of content: an image, a video, an HTML or generative content.*

**Remove or add a content, a mask, audio, 3D object or FX in the inspector:**

In the video or audio tab of the inspector of a clip container you will find all assigned assets

- **Drag** a mask, 3D object, audio or content **to the corresponding fields** in inspector to change them.
- Open the context-menu with a right-click to **clear or remove an asset**



## OTHER TYPES OF CLIP CONTAINER

## Devices

- Drag a device from the library into your sequence playback editor and VERTEX creates a clip container automatically.
- This is useful for devices that have animatable properties (like e.g. DMX devices have)
- **Learn more** about devices [here](#)

## Overwrite Surface and Output Settings

- You are able to create a clip container for each **surface and output**.
- With this kind of Clip Container you are able to **animate surface and output settings**
- **When this clip container is being played back**, it overwrites the global settings of an **output or an surface temporarily** as long as its duration allows.

*There are two ways to create such a clip container:*

1. **Drag** an output or a surface with your mouse **from the project explorer into your sequence**.
2. **Switch to programmer mode**. Change the settings of a surface or output in programmer mode. A clip container with keyframes is created when you store your programmer scene.



### **Check Clip Container Target**

Clip containers for surfaces and outputs always have an output as target - instead of a canvas.

Please double check if the correct output is set as target of your clip container.

## Global Layer and Console Layer

- **Settings for global layers can be overwritten** with clip containers as well.
- **Once the playhead runs into the clip container**, the settings of a **global layer are temporarily overwritten**.

*There are two ways to do this:*

1. **Drag a global layer** with your mouse **from the project explorer into your sequence**.
2. **Switch to programmer mode**. Change settings of a global layer in programmer mode. A clip container with keyframes is created when you store your programmer scene

## Notes

- A simple clip container that **hosts and shows your notes**.

- **Create a Clip Container for Notes:**
  - Go to the playback editor and open the context menu (right-click).
  - Select "Add"
  - Select "Add Notes Clip"

## SMPTE Clip

- generates SMPTE timecode when playhead is running into the clip container.
- For further information read the chapter for [SMPTE](#).

# WORKING WITH CLIP CONTAINERS:

## COMMON TASKS

- Copy/Paste a clip container and its settings (or keyframes) with shortcuts (CTRL + C / CTRL+V) or with help of the context menu.
- Double-click on a clip container to open the [Keyframe Editor](#)
- Move a clip container in the timeline or between tracks by simply dragging it.
- Adjust a Clip Container's length by edge editing: move the mouse over the edges at the beginning and end of a clip container and the arrow will change to the edge editing tool <--> automatically.
- Set a [User Color](#) for your clip container.

## SELECTION

There are various options to select clip containers in your Sequence Editor:

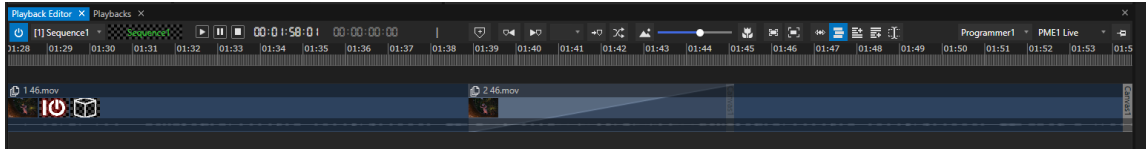
- Select multiple clips by dragging a lasso around them.
- or
- Right-click on a track or an empty space in the Sequence Editor to open the context menu and go to **Select...**

From there you can choose to select all clips vertically (at current playhead position), all upcoming/ previous clips and all clips on the current track.

A multi-selection allows property changes of all selected clip containers in the Inspector.

## AUTOFADE

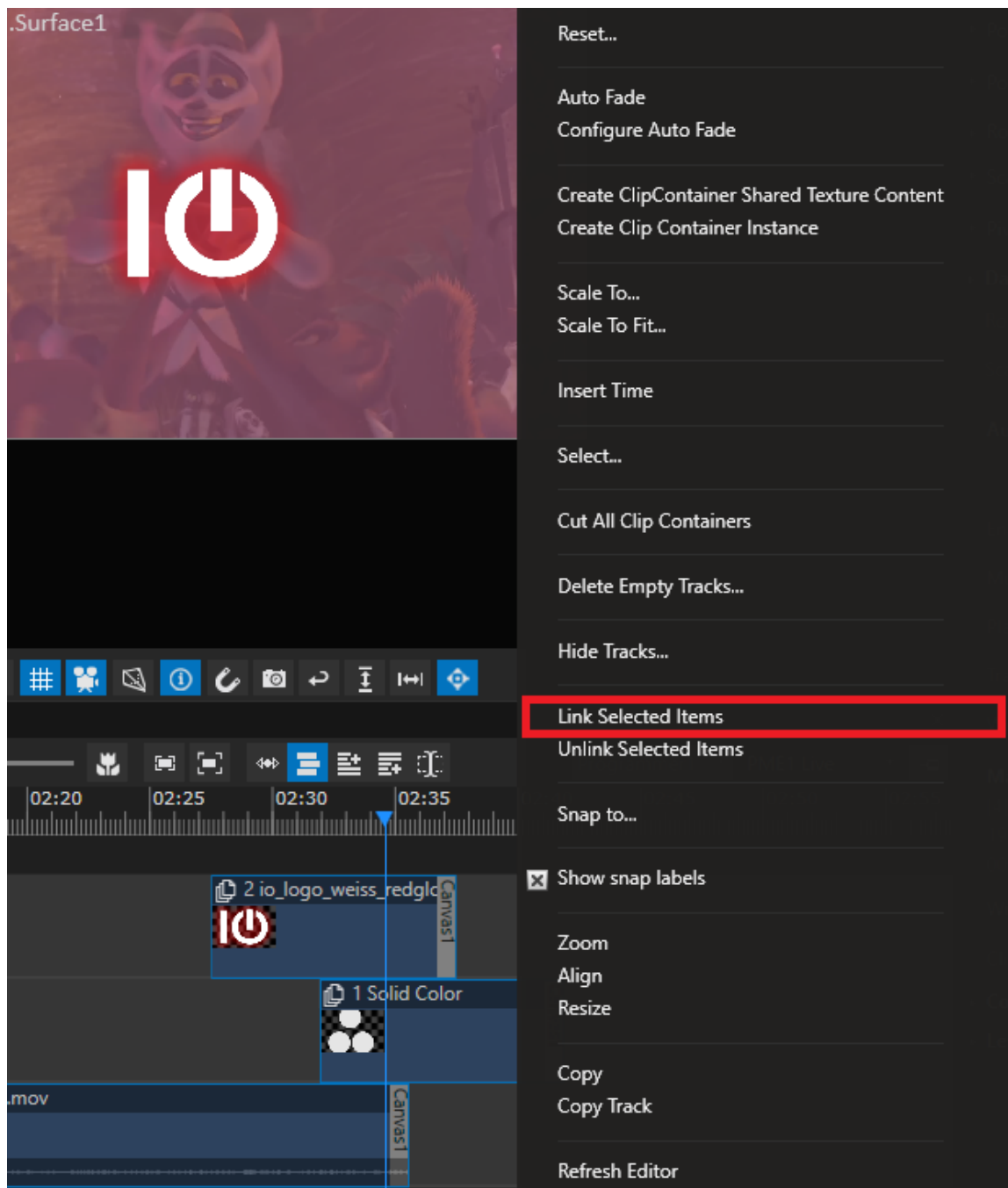
- When you drag a Clip Container into another one on the same Track, VERTEX automatically generates an auto-fade for video and audio.



## LINK AND UNLINK

- Link or unlink a selection of clip containers as a group.
- You can move linked clip containers "as a package"
- Property changes are taken over for all linked clip containers

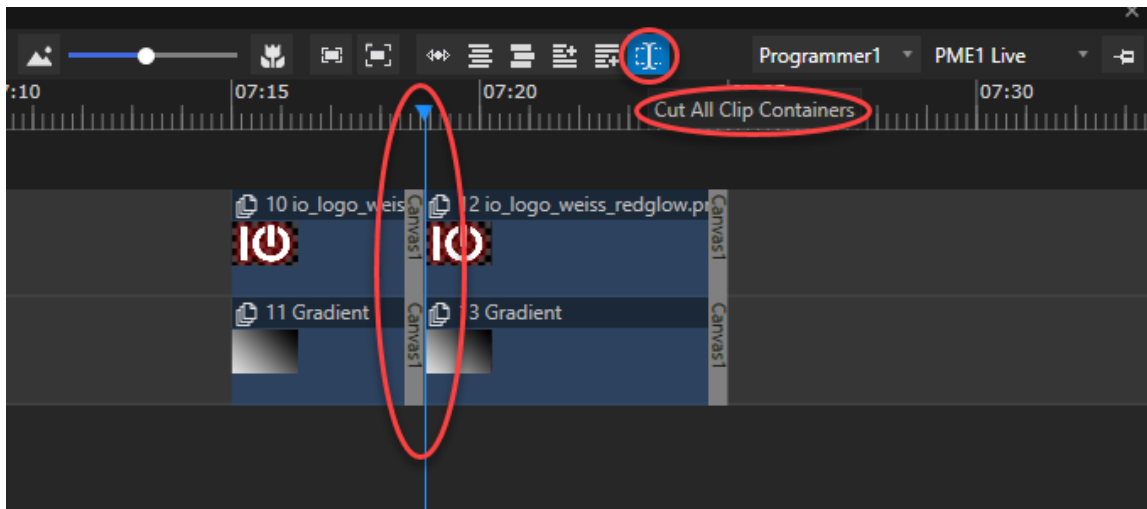




## CUT CLIP CONTAINER

- Splits a clip container into two parts at playhead position.
- You have 2 options:

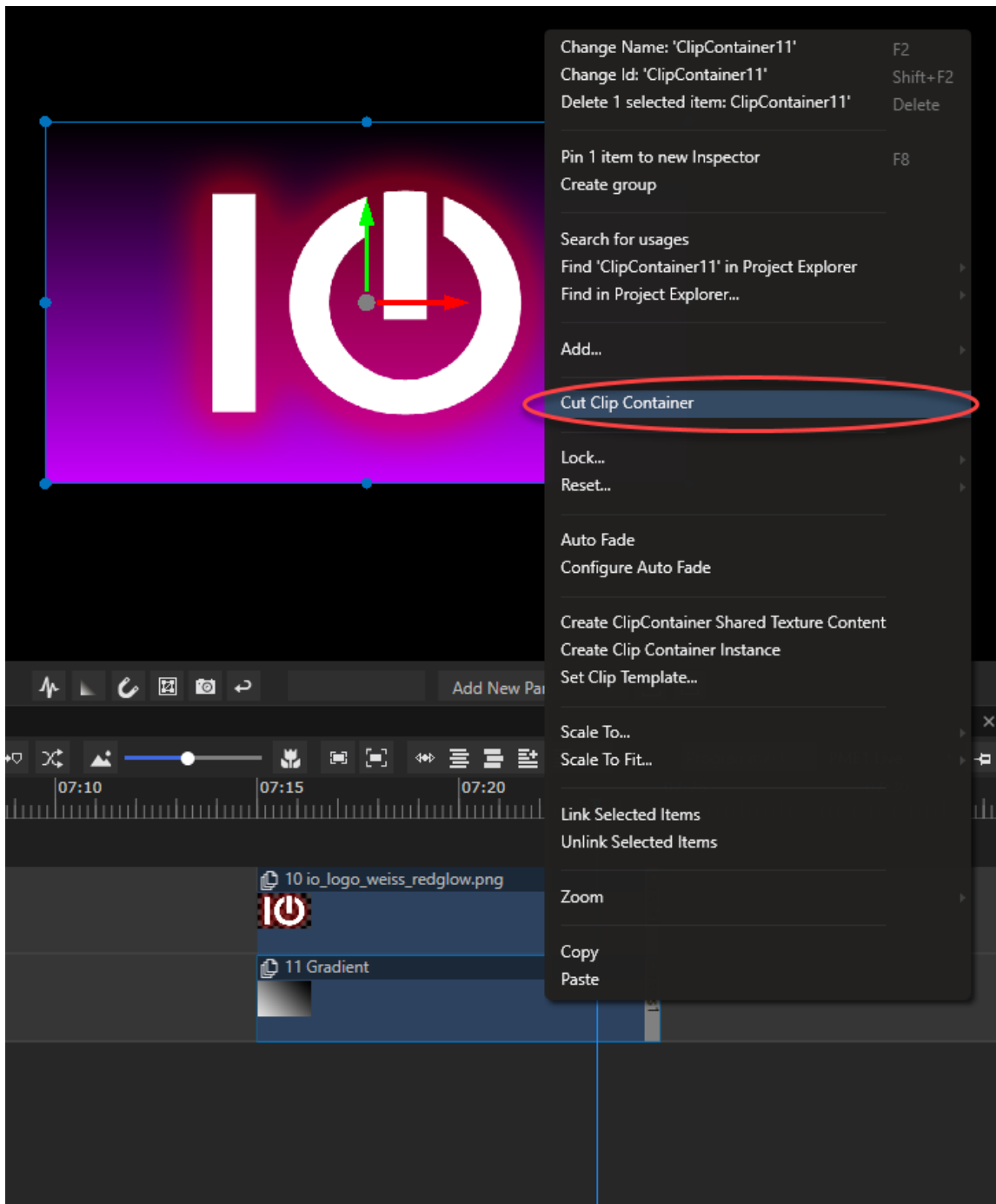
### **CUT ALL CLIP CONTAINERS AT PLAYHEAD POSITION**



- Use the cut-button in the Playback Editor
- or
- Select a clip container
- Open the context menu with a right-click
- Select "Cut All Clip Containers"

#### **CUT ONLY ONE CLIP CONTAINER AT PLAYHEAD POSITION**

- Select a clip container
- Open the context menu with a right-click
- Select "Cut Clip Container"



## SETTINGS

There are several properties you can set for a clip container.

Some of them depend on the type of clip container ( surface, content, device), whereas some of them are valid for every type.

	<p><i>Start Time</i> <i>End Time</i></p>	<p>Sets the points in the timeline where a clip container starts or stops. A reset is possible via the context menu. Select a clip container in your playback editor or project explorer and use the command "Reset to Cliplength"</p>
	<p><i>Length</i></p>	<p>Sets the clip container's length in the timeline. By default the length is set to the duration of the original content audio or video clip. Any other clip containers holding colors, images, textures or data are set to a length of 10 seconds by default. The clip container's length is relative to its start and end times.</p>
	<p><i>InPoint</i> <i>OutPoint</i></p>	<p>Sets the point of time within the content from which the clip container rolls in and out. For instance, if your clip is just 10 seconds long and the original content exceeds that time, you can select the frame which will be played in the clip. These Settings, however are not relative to the clip's length or start and end times. One of the many ways to create video loops in VERTEX.</p>
	<p><i>PreRoll</i> <i>PostRoll</i></p>	<p>Sets a buffer time for loading content before and after the playhead rolls in and out of a clip container.</p>
	<p><i>Enabled</i></p>	<p>Enable or disable a clip container to free your system's resources. Content is neither loaded nor rendered if the clip container is disabled (box is set unchecked).</p>
	<p><i>Freeze Property Updates</i></p>	<p>Freezing static non animated clip containers will reduce the amount of property updates per rendered frame. In doing so the properties are only updated when the playback enters a clip.</p>
	<p><i>Lock Clip To Time</i> <i>Lock Clip To Track</i></p>	<p>To prevent from accidentally moving, lock a clip container to time or to track. Both settings also are available in the context menu of a clip container. When set, a lock icon appears in the clip container.</p>
	<p><i>Pre-Loaded</i></p>	<p>This option will permanently load all textures and object data into into system RAM and GPU RAM. Please use only when required as this setting will fill up your system's resources.</p>

<i>Render In View Only</i>	<i>VERTEX's render engine processes everything on canvas where the playhead points in your timeline. Rendering of clip containers with this setting enabled will be bypassed if the clip is not in view of a surface. A clip container will be rendered when in view only.</i>
<i>Use In/Out Points</i>	<i>Enables the use of specific in- and out-points. See above.</i>
<i>Template Name</i>	<i>Name this clip for external references and watch-folder usage</i>
<i>Clip</i>	<i>Reference to the content of this clip container.</i>
<i>Clip Target</i>	<i>Sets the canvas where the clip will be displayed. To change it just drag another canvas from the project explorer and drop it into this field or into a clip in the playback editor.</i>
<i>Template Clip</i>	<i>Names the source of the template clip.</i>
<i>Fade In/Out</i>	<i>Sets the desired fade length if you choose to fade the clip in or out.</i>

## ICONS ON CLIP CONTAINERS

*When certain settings are activated, their respective icons appear on the Clip Container. Here is a quick overview:*



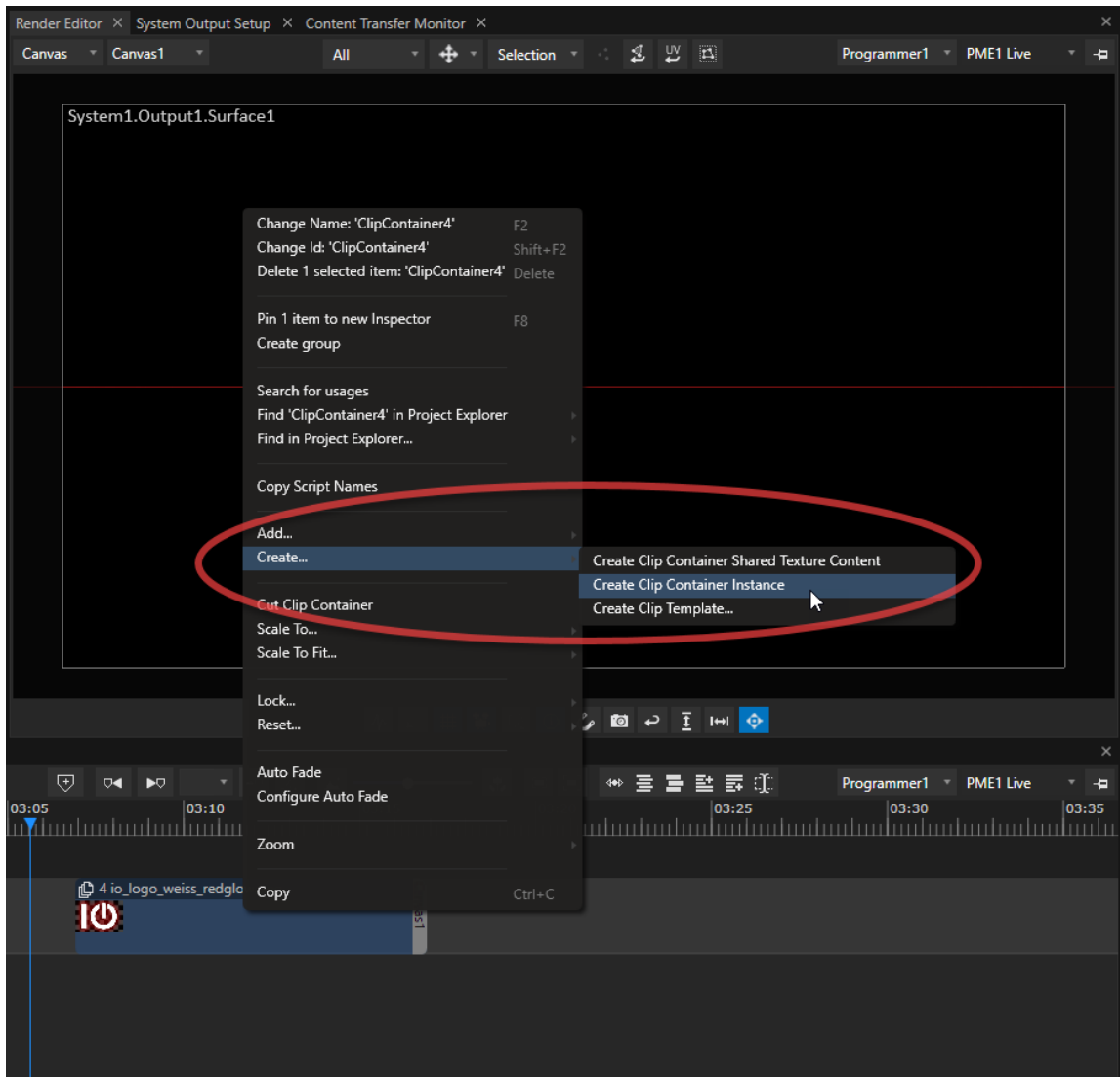
## Instances

- Instances offer an quick way to duplicate and link clip containers.
- All property settings including keyframe edits are duplicated and linked for all instances of a clip container.
- Clip container instances are marked by default in deep gray color.

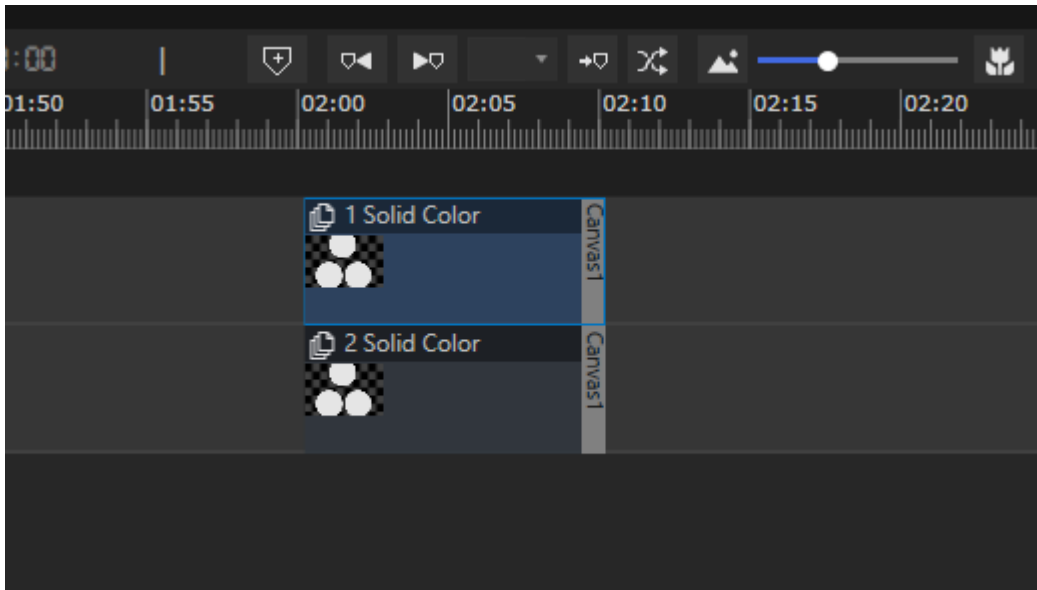
### Create Instance

- Select a clip container
- Right-click and open the context menu

- Select Create -> Create Clip Container Instance
- A new instance is created by default on a new track at same position.



- Work with the clip container instance and move it around independently.
- Property changes on the original container are taken over to all Instances.
- changes on an instance are taken over also to original clip container.



## Shared Textures

*Related to the instance function are clip containers with shared textures of content.*

*Creating a clip container that shares another clip's texture is **a resource friendly way to output the same content multiple times** simultaneously.*

*The main difference to creating an instance of a clip is that the clips' **property settings are not linked**.*

*Also, the clip with a shared texture will only be played out at the same time the original clip is being rendered.*

**Create** a shared texture **by selecting the source clip** with the content you want to share and go to **Create...** in its **context menu**.

**A new clip container** with a shared texture will appear in the Project Explorer's Content section and while **the source content is only loaded once** into the system's memory, it **can be used multiple times** in the sequence.





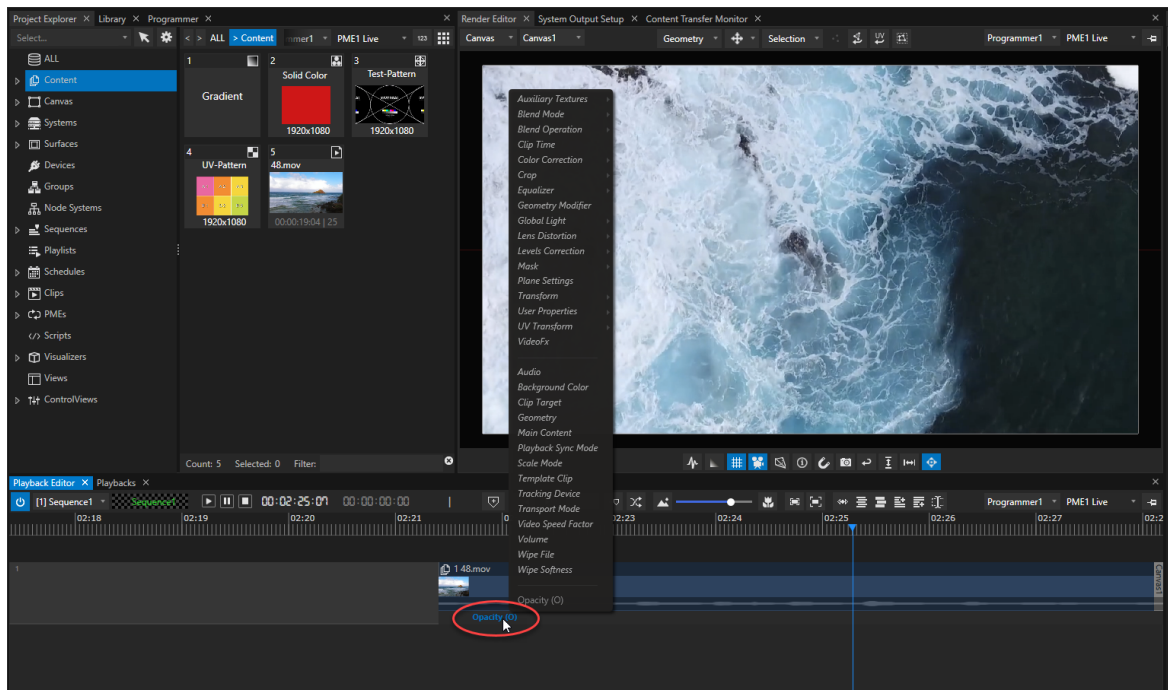
### 5.3.6 Keyframes

- By **placing keyframes** into a clip container, you can **animate changes in property values and parameters** at specific points in your sequence.
- VERTEX calculates a curve from keyframe to keyframe and interpolates the data. You can choose the type of curve from **linear, bezier** and **on/off switch**.
- **Keyframes are hosted by clip containers.** Their **position and time settings are relative and locked** to their clip container: Moving a clip container, moves all keyframes together along.

## Open The Keyframe Editor In The Playback Editor

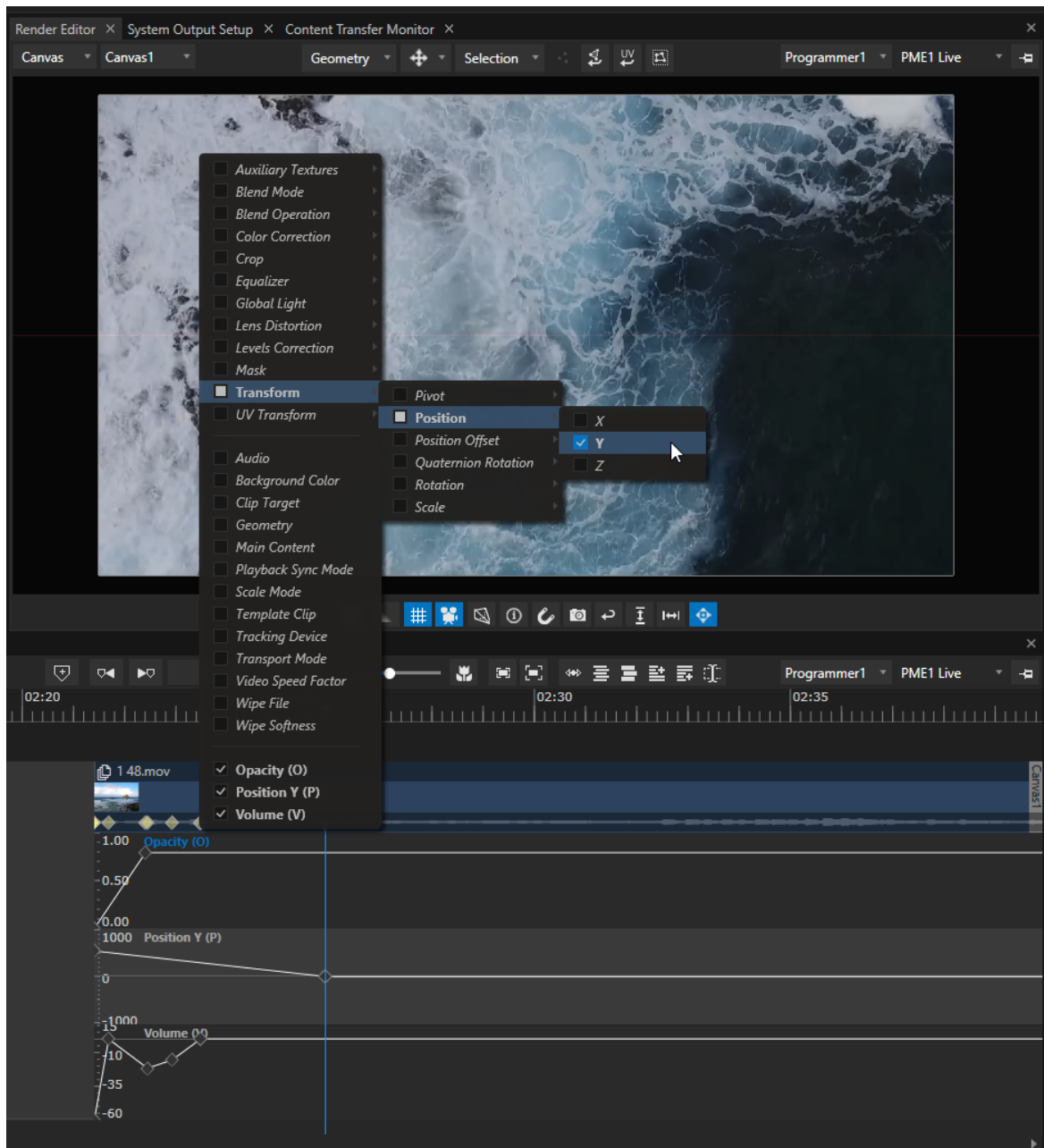
#### ACCESSING THE KEYFRAME EDITOR OF A CLIP CONTAINER:

- Double-click on a clip container to show/ hide its keyframe editor. It will appear below the clip container.
- The default property displayed in the new editor is **Opacity (0)**.



Click on this property to access a context menu with any other properties to choose from. Here you can filter the data displayed.

- CTRL+ click on the property name for a multiple selection. All selected properties will be shown in their own editor lanes.



The property name of the keyframe lane you're currently working in is always highlighted blue.

- Double-click next to the property name or anywhere in the lane to expand/ collapse the keyframe editor. You can also do this from the context menu (right-click).

**ADDING KEYFRAMES** is easy:

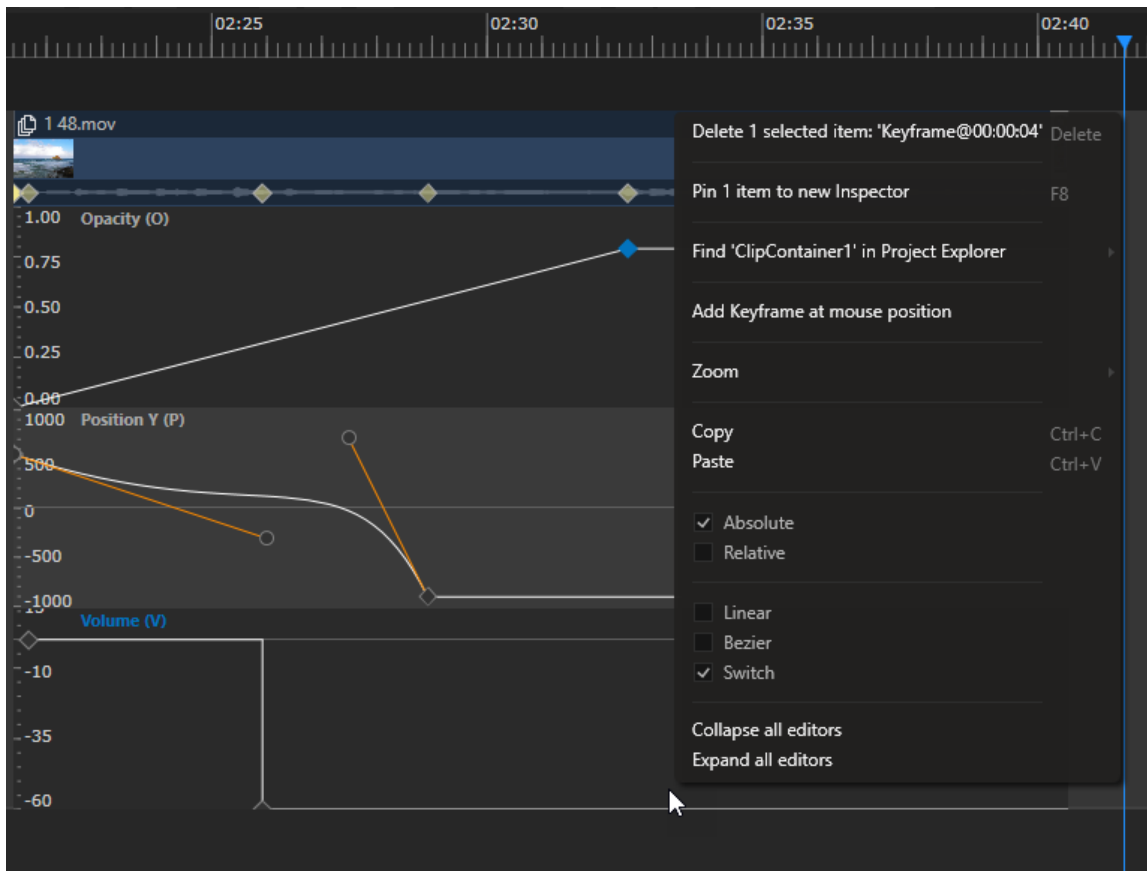
- Just **press K** in a property's editor lane and a keyframe will be added at the playhead's position.
- or
- **CTRL-click** in the lane on the position where you want to add the keyframe. If the editor lane is expanded, you can even add the keyframe with its **desired parameter value**.

- **Navigate** between keyframes with the shortcut *Shift + right or left ARROW KEY*.
- Double-click on a keyframe to **directly edit the keyframe's parameter value**. A small value window pops up next to the keyframe.
- **Zoom in and out on vertical axis** of the keyframe editor by holding down *CTRL* and using your mouse wheel.
- **Select one or more keyframes to copy / paste them**. They will be inserted at playhead position. This also works for pasting keyframes from one to another clip container.



*In value mode, after first keyframe has been set for a clip container, every value change in the inspector will automatically generate a new keyframe for the current property at the playhead's position. This can be done while playback is running and is a powerful tool to program animations and property changes on the fly.*

## Curve Types



There are **three types of curves** that interpolate the values between keyframes:

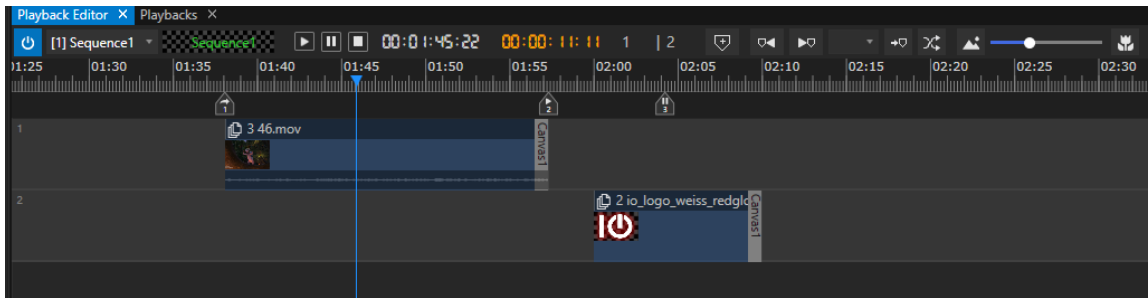
- **Linear** - just a straight line
- **Bezier** - a curve that can be shaped by two control points
- **Switch** - just binary on/ off behavior


You can set these types either in the inspector or in the context menu (right-click on a keyframe).

### 5.3.7 Cue

- **control and program your sequence** with the help of cues in the playback editor.
- a cue can be set to different **modes of functionality: continue, jump, pause or wait.**
- use **script commands** to program a more complex logic for a cue.

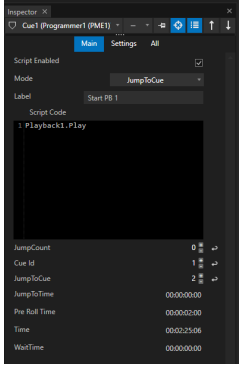
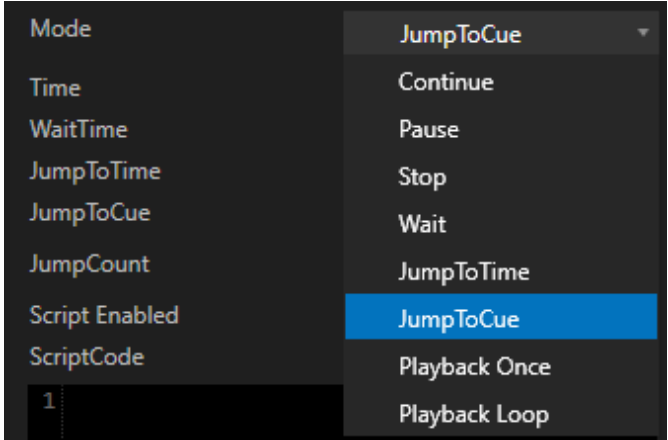
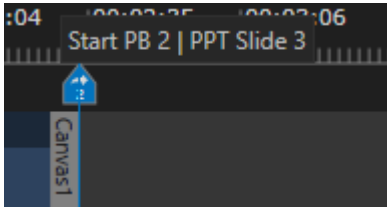
## User Interface



- Create a cue at the playheads position with the button  in the playback editor or use "Add" from the Context Menu (open with a right-click in the playback editor).
- The orange timecode displays the remaining time until the next cue.
- Navigate through cues with the help of the buttons in the top bar of the [playback editor](#).
- Select a cue and set its mode or other settings in the inspector.

## Settings

	<b>Script</b> <b>Enabled</b>	<i>Default: Enabled</i> <i>Uncheck to disable script execution for this cue.</i>
	<b>Mode</b>	<i>Sets mode and behavior of a Cue</i>

		 <p><b>Mode</b> JumpToCue</p> <p>Time Continue</p> <p>WaitTime Pause</p> <p>JumpToTime Stop</p> <p>JumpToCue Wait</p> <p>JumpCount JumpToTime</p> <p>Script Enabled <b>JumpToCue</b></p> <p>ScriptCode Playback Once</p> <p>1 Playback Loop</p> <p><b>Default Setting:</b> Continue continuous playback when cue is reached by playhead.</p> <p>Options:</p> <p><b>Pause:</b> pauses playback when cue is reached by playhead.</p> <p><b>Stop:</b> stops playback when cue is reached by playhead.</p> <p><b>Wait:</b> Playback will wait at the cue for a time that is defined as <b>WaitTime</b>. When the wait time passes by, the timecode display in the playback editor changes to red color.</p> <p><b>JumpToTime:</b> Jumps to a defined time when cue is reached by playhead. This time has to be defined in the <b>JumpToTime</b> property. By default this time is 00:00:00:00.</p>
	<p><b>Label</b></p>	<p>Label your cue with text. This label also is shown as tooltip in playback editor when run over with your mouse cursor.</p> 
	<p><b>ScriptCode</b></p>	<p>Enter script commands that will be executed when the cue is reached.</p> <p>Click into the script terminal and use the shortcut" CTRL + Space" to show all available commands.</p>
	<p><b>JumpCount</b></p>	<p>Parameter for modes <b>JumpToCue</b> and <b>JumpToTime</b>. You can set the number of times your jump command needs to be executed. After the number is reached, the playback continues</p>

		<i>from the cues position and the jump is no longer executed. The default setting is zero and means the jump will proceed indefinitely.</i>
	<b>Cue ID</b>	<b>ID of a Cue.</b> <i>Essential identifier for a cue. Must have a unique number for each sequence and appears on each cue in the playback editor.</i>
	<b>JumpToCue</b>	<i>Set the cue ID of the cue you want to jump to.</i>
	<b>JumpToTime</b>	<i>Defines the jump's target time when cue mode is set to <b>JumpToTime</b>.</i>
	<b>PreRoll Time</b>	<i>When in jump mode, you can set the time of pre-load for the next cue.</i>
	<b>Time</b>	<i>Position of the cue in the timeline.</i>
	<b>WaitTime</b>	<i>Time that the playback waits when cue mode is set to <b>Wait</b>.</i>



**Changing the sequence framerate may shift already existing cues!**

*When changing the frame rate of a sequence that already has cues in the timeline, all cues will be shifted to the next valid time code frame.*

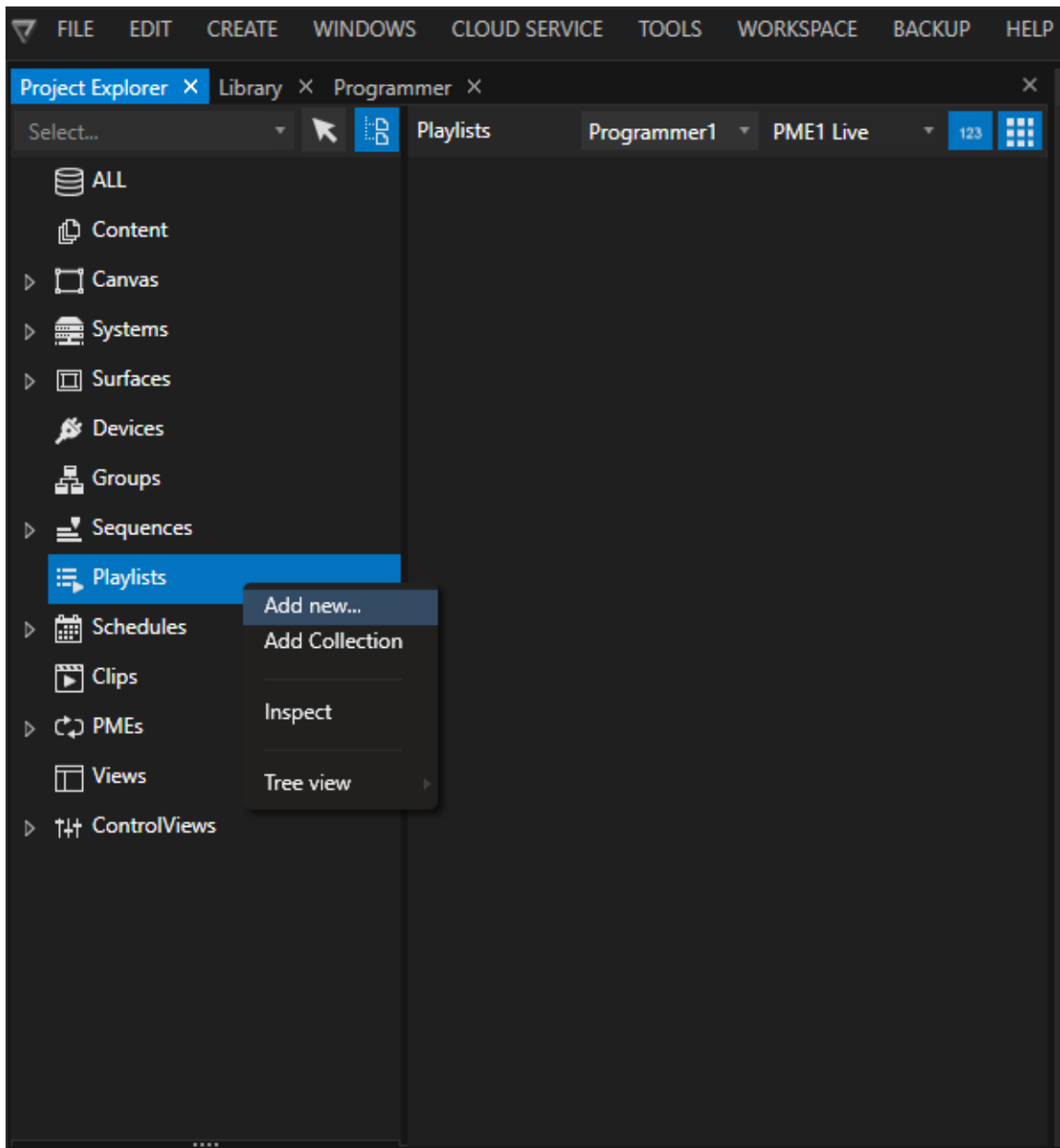
*Depending on your changes, this recalculation may cause a minimal shift in cues - please inspect all your cue times after changing the framerate.*

### 5.3.8 Playlist

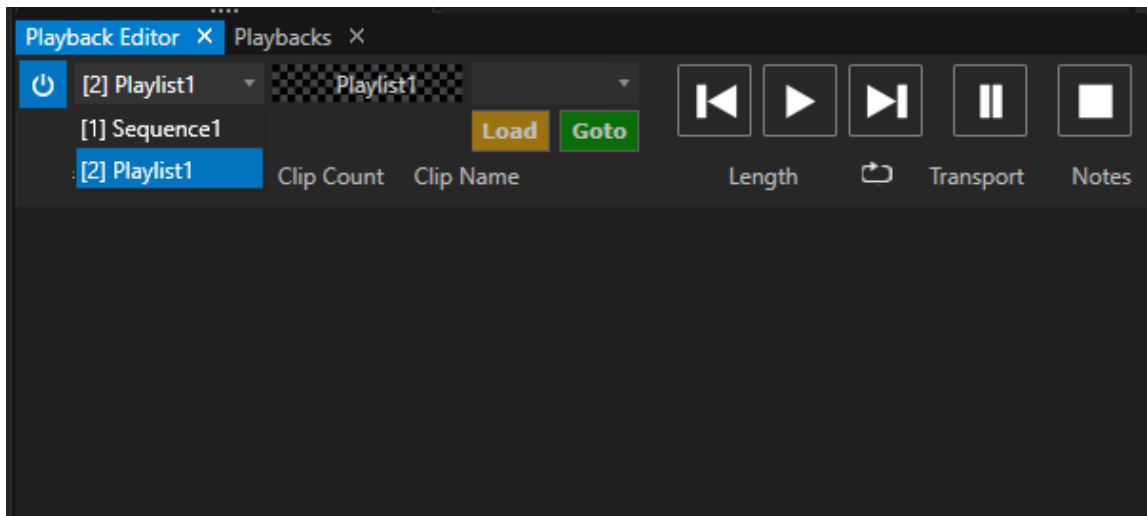
- A Playlist is a **list based Playback Provider** that is **hosted into a Playback**
- A Playlist **contains a list of Clips, sorted on different Tracks** into a list.
- Compared to a sequence, the **Cue number** of a Playlist is the **same as the Track number**
- A Playlist has a **default Canvas**, but every Clip could be assigned to a different Canvas or even output

## Create a Playlist





- On Startup, VERTEX creates a timeline based [Sequence](#) for you.
- To create a new Playlist, just navigate to the Playback section into [Project Explorer](#), open context menu by a right click and select "Add New"
- Or just use "Create" in the main menu on the top bar
- Go to the [Playback Editor](#) and switch to the new [Playback](#) "Playlist 1"



## User Interface and Workflow

The Playlist in VERTEX is **ready for a quick start with its default settings**.

For special needs, **different options and settings** gives you a **powerful and flexible way to handle also more complex tasks** with a Playlist.

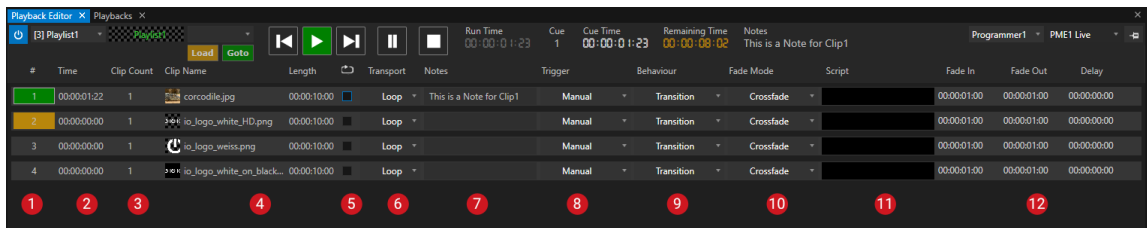
- Drag Content from Project Explorer to the Playback Editor. For each Content item a new Playlist Track is created



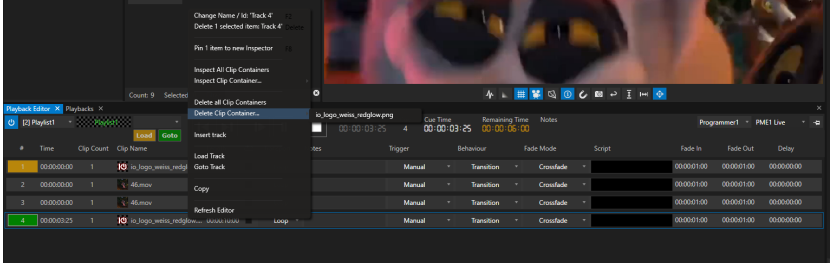
### Sequence as Playlist Track

It is also possible to drag a whole Sequence as Content into a Playlist.

- To change the track order:
  - Select one or multiple Tracks, hold the left mouse button and drag it to a new position
  - Select a track, go to Inspector and change "Cue Id"



<p><b>1</b></p>	<p><b>Playlist Cue Number</b></p>	<p>Shows the Cue/Track number</p> <p><b>Status colors</b></p> <p>Yellow: Clip is preloaded into cache</p> <p>Green: Currently Playing</p>
<p><b>2</b></p>	<p><b>Playback Time</b></p>	<p>Shows the playback time of the Clip/of the Clips that is/are currently playing</p>
<p><b>3</b></p>	<p><b>Clip Count</b></p>	<p>Shows the amount of Clips on a Track</p> <p><b>Default:</b> only 1 - when <u>Track Clip Mode</u> of a Playlist is set to Single Clip default setting)</p> <p><b>Option:</b></p> <p>If you want to work with more than one Clip on a Track, set <u>Track Clip Mode</u> of a Playlist to Multi Clip.</p> <p>The Clip number shows you the number of Clips on a Track</p>
<p><b>4</b></p>	<p><b>Clip Name</b></p> <p><b>Clip Length</b></p>	<p>Shows Clip Name, and thumbnail of the first Clip of a Cue</p> <p>Shows Clip length of the first Clip of a Cue</p> <p><b>Inspect Clips with the help of the context menu. Right-click with your mouse on Track.</b></p> <p>Choose "Inspect Clip Container..." to select one of the Clips into Inspector</p>

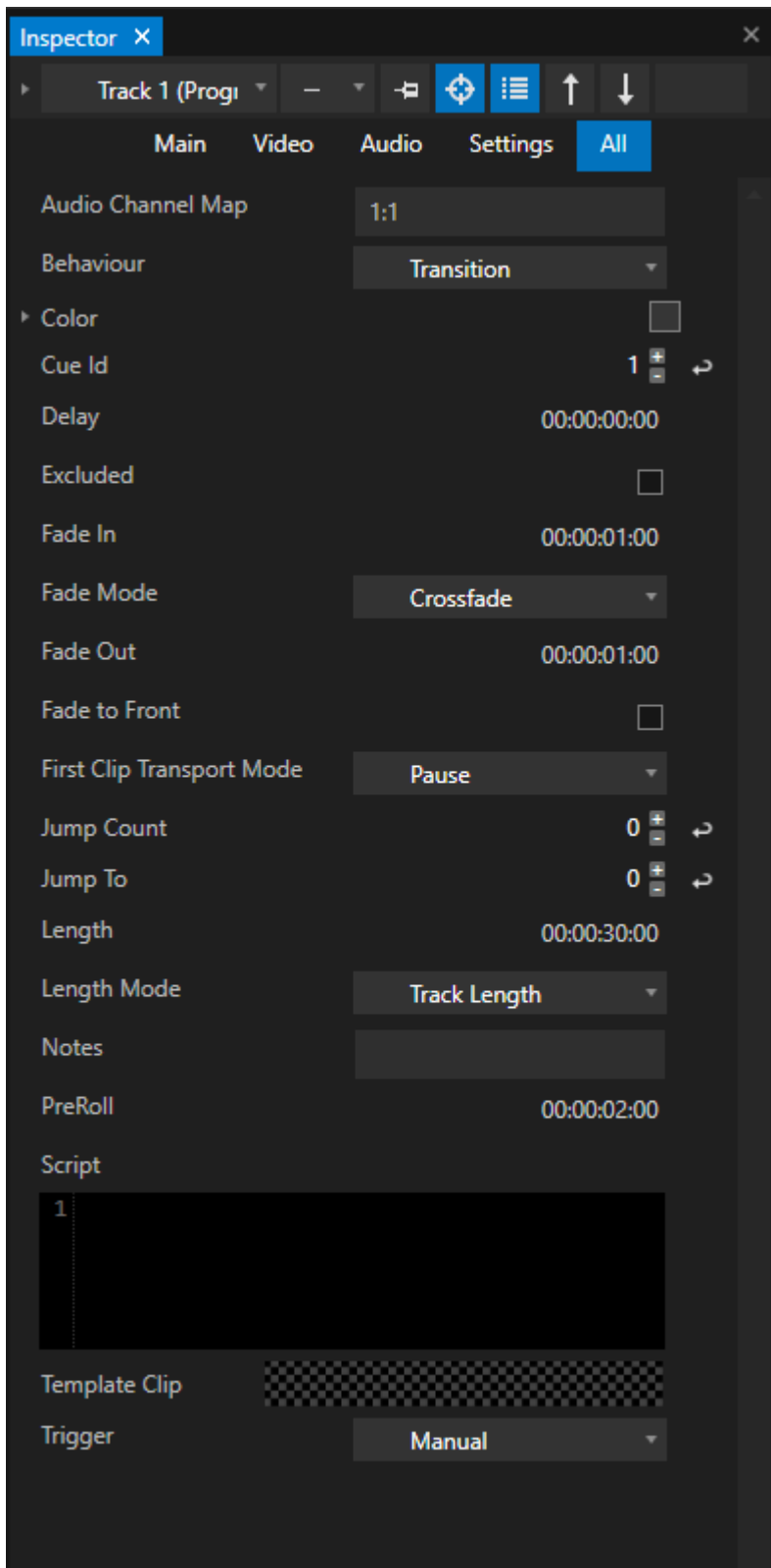
		<p>Choose "Delete Clip Container..." to remove one of the Clips</p> 
5	<b>Freerun</b>	<p>Sets Freerun on/off for a video Clip</p> <p>Default: off</p> <p>When set to off, the Clip Playback is synced to the System Clock</p>
6	<b>Transport Mode</b>	<p>Sets Transport Mode for a Clip. relevant when Track length of a Playlist Cue is longer than the length of the video clip or if Freerun is enabled</p> <p><b>Options:</b></p> <p>Loop (default) - Loops Video Content is Track length of a Playlist Cue is longer than the lengths of the video clip</p> <p>Pause: Just pauses the Clip</p> <p>Once: Just play once</p>
7	<b>Notes</b>	<p>Add a Note.</p>
8	<b>Trigger</b>	<p>Sets method how this Cue will be started /triggered</p> <p><b>Follow (Default):</b> Follows the previous Cue. Cue is triggered after previous Cue is finished</p> <p><b>Manual</b> Manuel trigger is necessary to start the Cue. Cue will not automatically starts after previous Cue is finished A manual trigger could be the Play Button or even a Script Command</p> <p><b>With Previous</b> Cue will be triggered with previous Cue. When previous Cue is started and played, Playback for this Cue also will start.</p>

<p><b>9</b></p>	<p><b>Behaviour</b></p>	<p>Defines the Cue behaviour at its end.</p> <p><b>Transition (default)</b>  <i>Transition is done between this Cue and the next upcoming Cue.  Both Cues are loaded into cache and running in parallel during transition time</i></p> <p><b>Release</b>  <i>Releases this Cue when next Cue is started.  Contrary to "Transition" both Cues are not loaded and running in parallel for the transition time.  Use e.g. for systems with less performance</i></p> <p><b>Hold</b>  <i>Hold the Content form this Cue also when next Cue already is started</i></p>
<p><b>10</b></p>	<p><b>Fade Mode</b></p>	<p>Defines the type of transition to the next Cue</p> <p><b>Crossfade (default)</b>  <i>Crossfade is processed between Cues</i></p> <p><b>No Fade</b>  <i>No Fade between Cues</i></p> <p><b>Fade In/Out</b>  <i>Fade in and/or fade out is processed between cues.  Timing could be set separately for in and out into Time Settings</i></p>
<p><b>11</b></p>	<p><b>Script</b></p>	<p>Enter a Script Command that should be executed when Cue starts</p>
<p><b>12</b></p>	<p><b>Time settings for Fade and Start</b></p>	<p><b>Fade In</b>  <i>Sets duration for a fade in. Is used when behaviour was set to transition</i></p> <p><b>Fade Out</b>  <i>Sets duration for a fade out. Is used when behaviour was set to transition</i></p> <p><b>Delay</b>  <i>Sets a Delay for Start</i></p>

# Settings

For a Track

*All Settings also could be done into Inspector  
Just focus a Playlist track with your Mouse.*







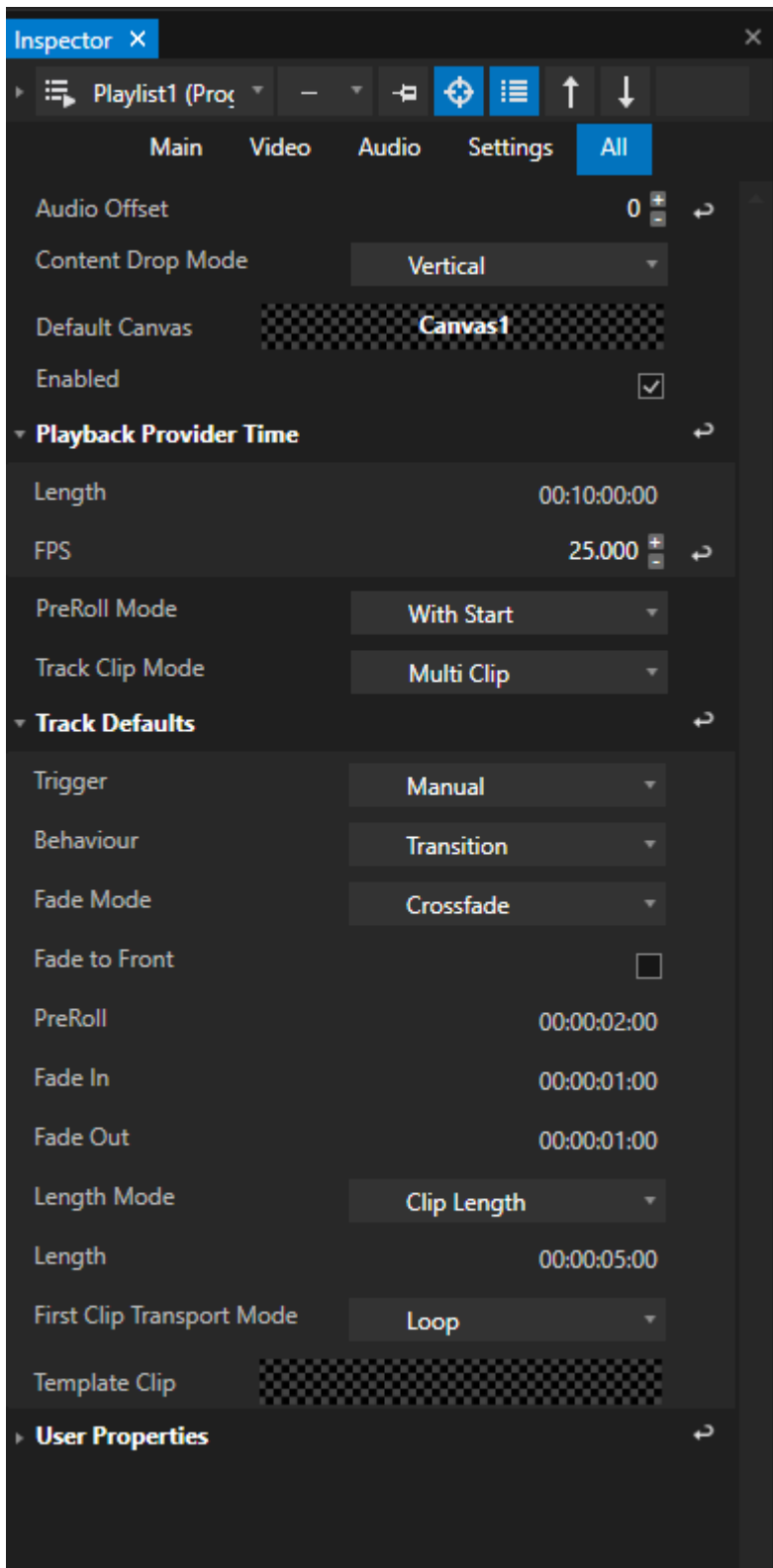
Please keep in mind that there is a **difference between this Track settings** with behaviour and transitions **and the settings for a Clip into this Track**

**To select a Clip into Inspector** and do settings like position in Canvas, corrections, etc, right-click on an Playlist entry, open the context menu and click to "Inspect Clip Containers..."

## For a Playlist

Similar to a Sequence, there are also some **settings for a Playlist**.

**Select a Playlist into Project Explorer** and change the settings into the **Inspector**



## Default Canvas

*Defines the default Canvas that is assigned to each Clip.*

*Drag with your mouse a Canvas from Project Explorer to this property field to change.*

## Audio Offset

*Set an Audio offset for the Playback of the whole Playlist.*

*Values are in milliseconds.*

*Enter a negative value to play Audio earlier than video*

*Enter a positive value to delay Audio from video*

## Length and FPS

*Changing FPS will affect the displayed Timecode for a Playlist.*

*Length will affect the length of the Playlist as Playback provider*

## Global Position Offset and Global Position Scale (Advanced Inspector Mode)

*Sets a Global Offset for Position for all Content into a Playlist*

*or*

*Scales all Content into a Playlist globally*

## Track Clip Mode

### **Single Clip (default)**

*A Track/cue of a Playlist can host only 1 Clip*

### **Multi Clip**

*A Track/cue of a Playlist can host multiple Clips.*

*Read Subchapter ["Multi Clip Mode"](#) for more information*

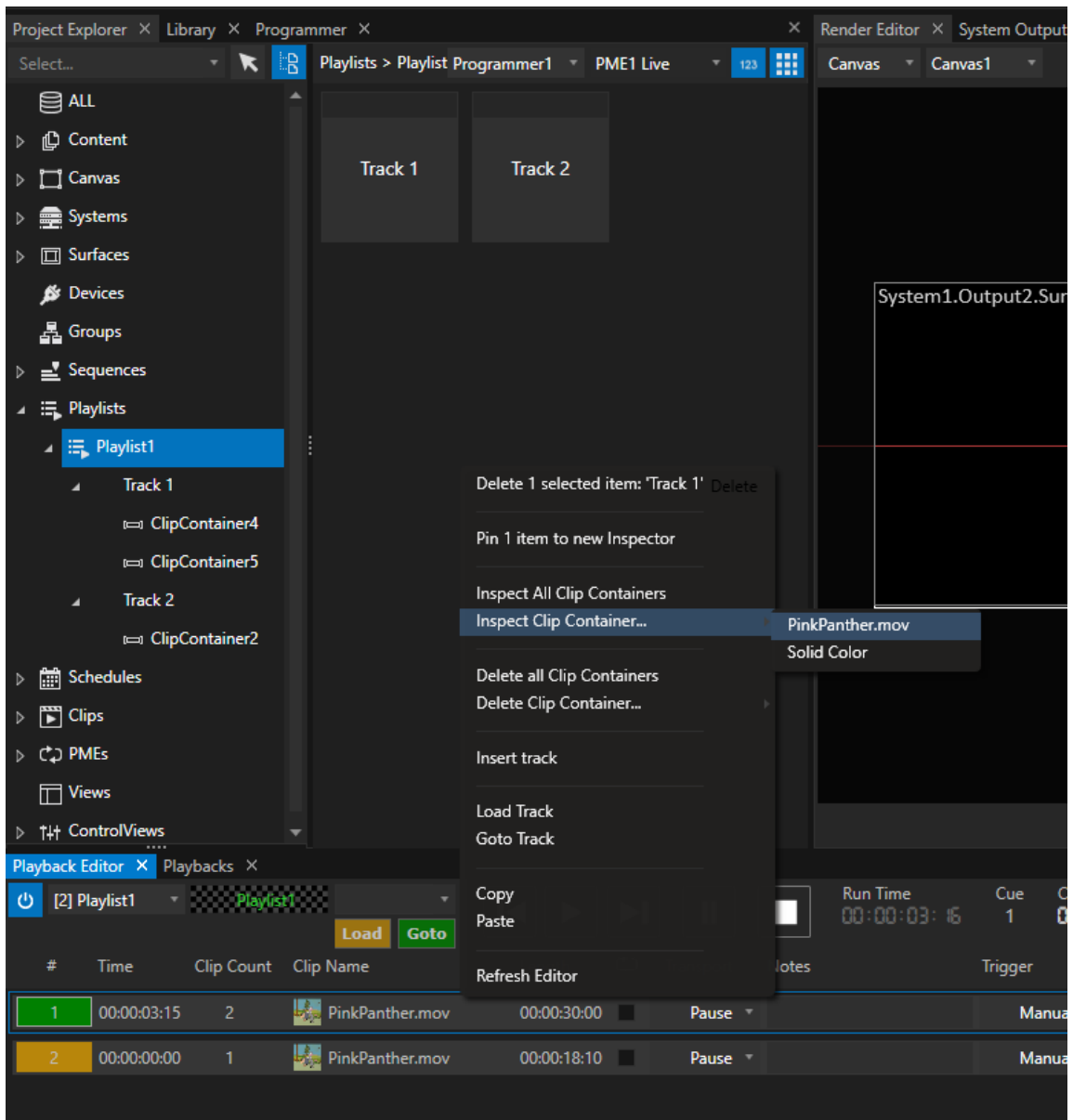
## Track Defaults

*Set up default values and settings that are valid for every new Playlist entry you will create*

*Read more about this setting in the [subchapter above](#)*

## Multi Clip Mode

- In "**Multi Clip**" Mode, VERTEX is able to host a **number of Clips on the same Playlist Track**
- With this feature you are allowed to handle more complex scenarios within a Playlist:  
e.g. to play different Clips on different Canvas with one single Playlist Cue.
- To use this feature, **you first have to change the Track Clip Mode of a Playlist**. By default, this mode is set to Single



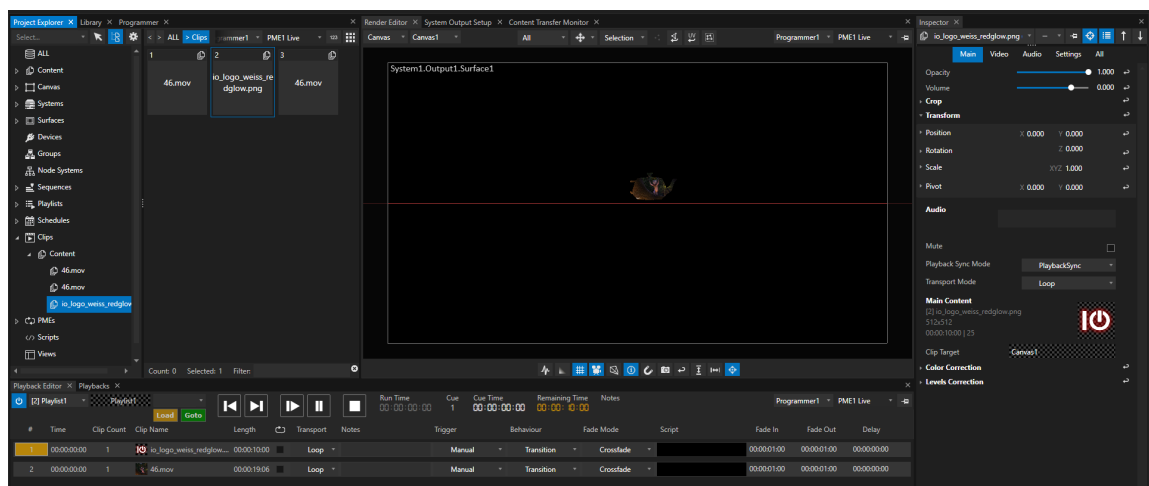
## Working in Multi Clip Mode

- Create a Clip by **first dragging just 1 content item** from Project Explorer into the Playback Editor
- Then, **drag one or more Content Items** from Project Explorer **to the same Track**: the Clip Count of this Track increases
- Also in Project Explorer all Clips now are listed **as child elements of a Track**
- **Inspect a Clip Container with help of the Context Menu**: Right-Click on a Track to open a list of Clip Containers or to Inspect All

- **Delete a Clip Container** with help of the Context Menu: Right-Click on a Track to open a list of Clip Containers or to Delete All

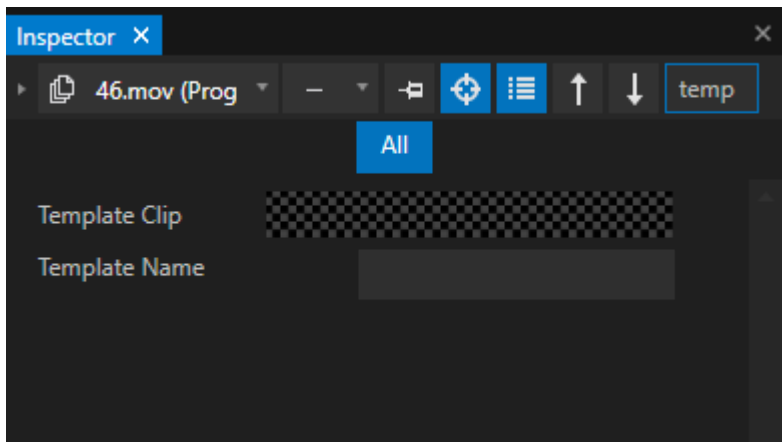
### 5.3.9 Clip

- Clips are created automatically by VERTEX when you add a Content from the Project Explorer to a Playlist or even a Clip Container into a Sequence
- All Clips are listed into the Clips Section in the Project Explorer
- Clips can be used as template for other clips



## Template Clip

- **Every Clip** can either be a template for other Clips
- A Clip which has a template assigned will inherit the properties and keyframe animations from the template clip

**Template Clip:**

Drag & Drop target onto which clips can be dragged from the "Clip" tree in the project explorer.

**Template Name:**

To use a clip as a template, enter a name for the template into this field.

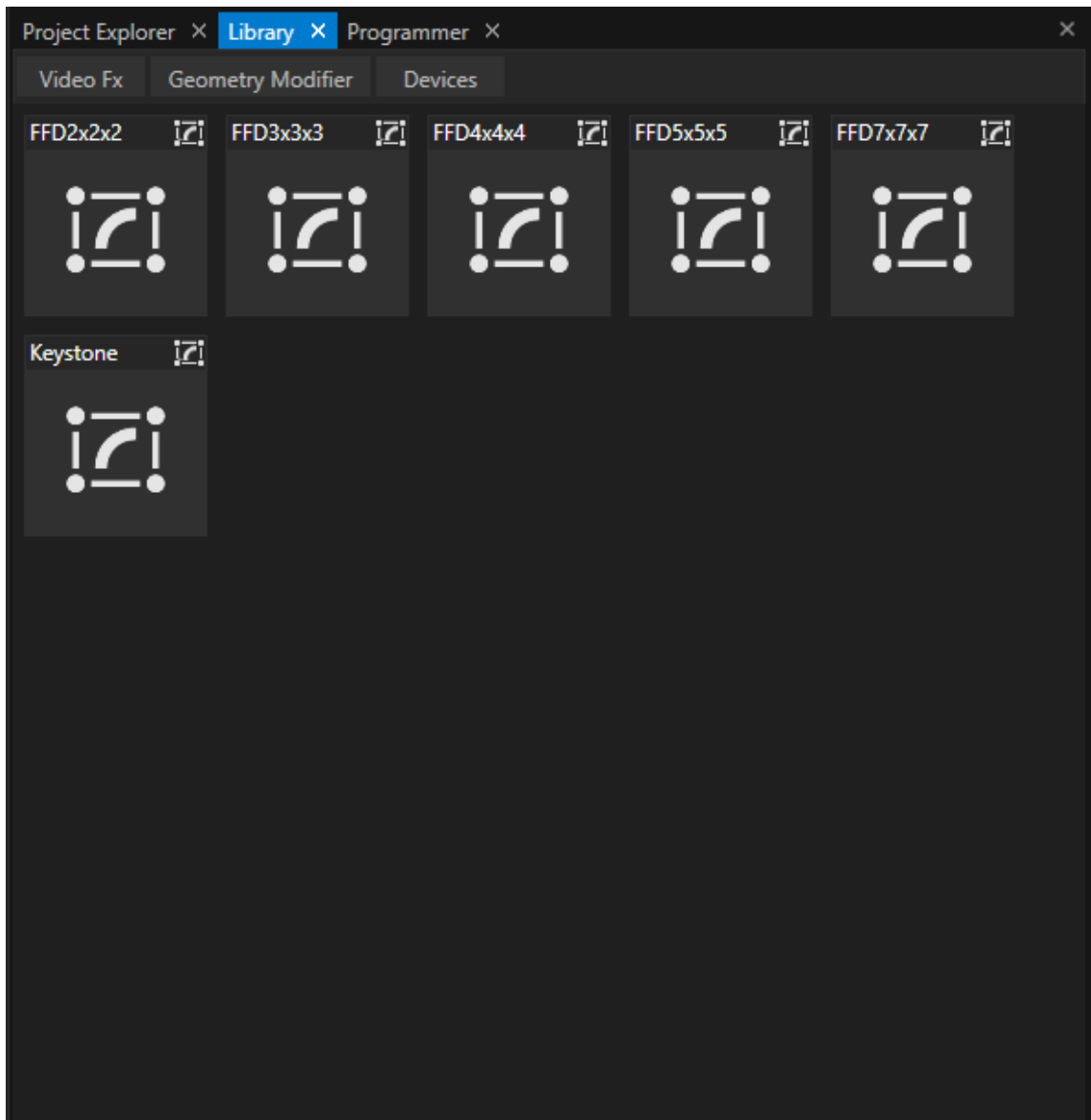
**Template Clips for WatchFolders**

[WatchFolders](#) gives you the option to automatically generate a Playlist or a Sequence based on Content files into a specified folder.

You have to define a Clip template first. Based on this template, your Playlists or Sequences will be generated

### 5.3.10 Geometry Modifiers

- Geometry Modifiers **help you to manipulate Planes or 3D Objects**
- **FFDs (Free Form Deformers)** help you to **easily manipulate a group of Vertices** to create e.g. Warpings
- Drag them from the [Library](#) to a Clip Container or to a Surface to use it



### 5.3.11 Video FX

- VERTEX comes with **library of video effects** that are easily applied
- **Drag any Video FX** from the [Library](#) onto a [Clip Container](#). The effect's settings can be adjusted in the Inspector.
- **Video FX can only be applied to a texture already in place:** the target of the video effect needs to be a clip container hosting either a video, an image or some generative pattern underneath the superimposed video effect. The texture of a Surface can also be used as a matte for video effects.
- **Multiple video effects** can be stacked and combined.





Video FX "Lens Flare" dragged on a Clip Container with a Solid Color. Settings for Video FX into Inspector are highlighted.

### 5.3.12 Audio

- VERTEX differs between **Preview Audio** for e.g. **Preprogramming** and **Live Audio** for final **Playout**
- **Volumes** can be set for a **Clip Container/Clip** or for a whole **System**
- There are different options to **route Audio Channels** and do **Audio settings** for a **Clip Container** or a **Clip**

## Preview and Live Audio

There is a main difference between **preview audio** and **live audio**:

### Preview Audio

- **Preview audio** plays out all audio from all playback mixing engines - That includes all playbacks in **PME live** and all playbacks in one (or maybe more) preview **PMEs**.
- **Preview audio** is played out from every system in your project.

## Live Audio

- Live audio plays out only audio from playbacks that are running in PME Live.
- Live audio is played out by an defined audio system that could be set for a canvas or a whole system.

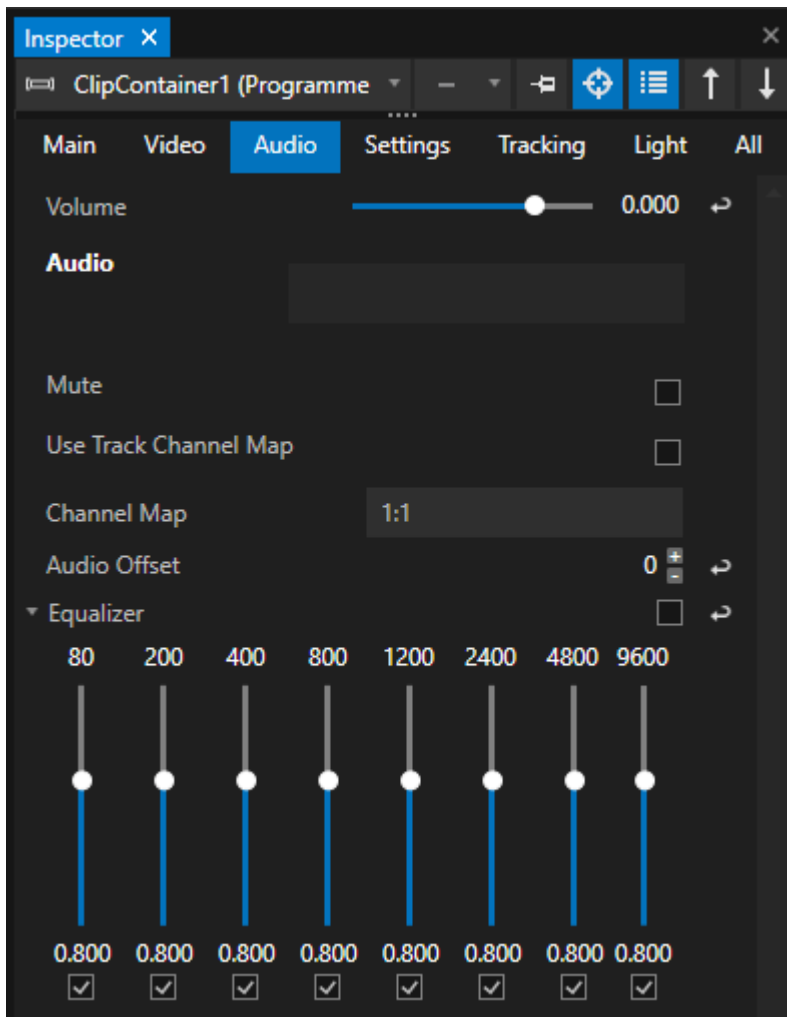
By default live audio is played out for all canvas on the same system. It is possible to define a certain audio system for a canvas.

For **more information about Audio Card /Audio Interface Settings** and **Audio properties and settings on Vertex System level**, please also read the [Audio Out](#) and [Audio Playback Chapters](#)

## Audio Settings for a Clip Container or a Clip

For a **Clip Container in a Sequence** or a **Clip in a Playlist** there are different options to **set or route Audio**.

To see all available Audio settings for an item, select the **"Audio"-Tab in the Inspector**



## Add or Exchange Audio

When working with a **Video that has embedded Audio Tracks**, VERTEX first always uses the Audio that comes with your Video.

In this case the **Audio Property field is empty**.

When you want

- to **exchange an Audio track for a Video**
- **or**
- to **add an Audio track** to a Video that has no Audio thus far

**then**

Drag an Audio Content from Project Explorer to the Audio Property field. Clear Audio via the Context-menu and a right-click with your mouse.

## Volume

Sets the **Volume for a Clip Container/Clip**.

You are able to set individual volumes for each ClipContainer/Clip

## Channel Routing

With an easy syntax you have access to the **VERTEX Audio Matrix** and are able to **route Audio Channels of a ClipContainer/Clip to Output Channels** of your Audio Output Card/ Audio Interface.

By default the **Audio Mapping is set to 1:1**

For a Video with a Stereo Audio Track this means:

Channel 1 is routed to Channel 1 of the Output.

Channel 2 is routed to Channel 1 of the Output.

**You are able to change this routing.**

When you want to route Channel 1 of your Video to Audio Output Channel 7 and Channel 2 to Output Channel 8, the routing syntax is:

```
1@7, 2@8
```

The **Syntax for an Audio Routing is**

```
AudioChannel#@OutputChannel#, AudioChannel#@OutputChannel#, AudioChannel1
```

To **reset a routing**, just type in

```
1:1
```

again.

## Use Track Channel Map

For a [Track of a Sequence](#) you also can create a **Channel Map**

When the **"Use Track Channel map"** check-mark **is set** for a Clip Container, this **Clip Container uses the Track Channel Routing**

When check-mark is not set (default), the Clip Container routing is taken into account.

## Audio Offset

With help of the Audio Offset parameter you can play the **audio track of a Clip Container/Clip** earlier or delayed to a video

Values are in millisecond.

To play Audio earlier: add negative values

to play Audio later than Video: add positive values

## Follow up Chapters for Audio

[Configure Audio Outputs](#)

[Audio Playback](#)

### 5.3.13 Devices

- Control **different types of external devices** directly from VERTEX
- Select a device from **library** and add it to your project
- Set device parameters with **script commands**

## Workflow

VERTEX comes up with an **abstract device model** that enables you to **control different types of devices with various functionality**.

Some of the device properties are animatable via keyframes like the ones of DMX-Devices for instance.

But basically all device parameters are accessible with VERTEX script commands.

**Regardless of the device type and its range of parameters, the basic workflow is always the same:**

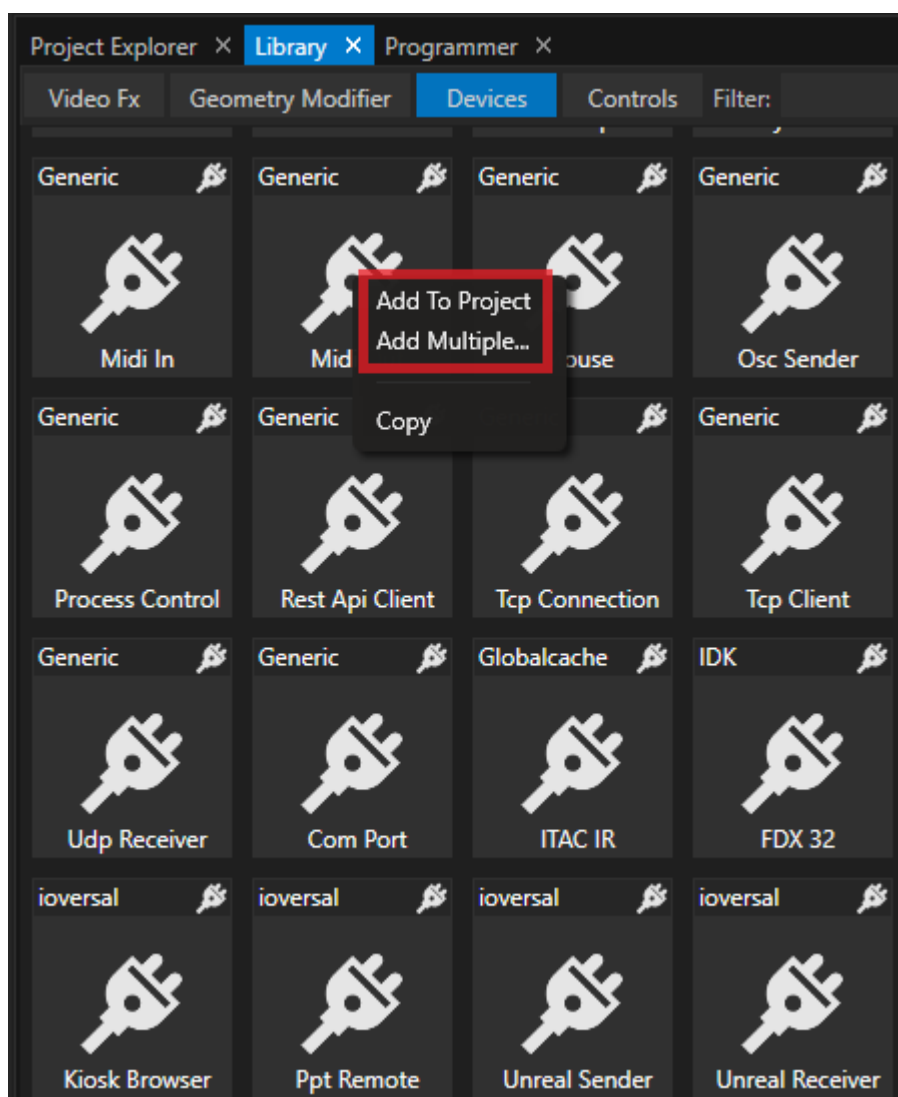
1. [Open the library editor and search for a device](#)
2. [Add the device to your project](#)
3. [Select device in the project explorer](#)
4. [Do connectivity and protocol settings in Inspector](#)
5. [Access device parameters with vertex script commands](#)

or

## 6. use a device in a clip container

Troubleshooting: Reconnect a Device

## Devices in Library Editor



- Open the **library**.
- **Search for the device** you want to use  
use the search filter and search by names
- **Right-click with your mouse** and select whether you want to **add only one or multiple devices** to your project

**Directory and Folder for Devices**

You are able to write your own device templates and add them to the VERTEX library.

You have to save your custom devices with the file extension `.vxd`

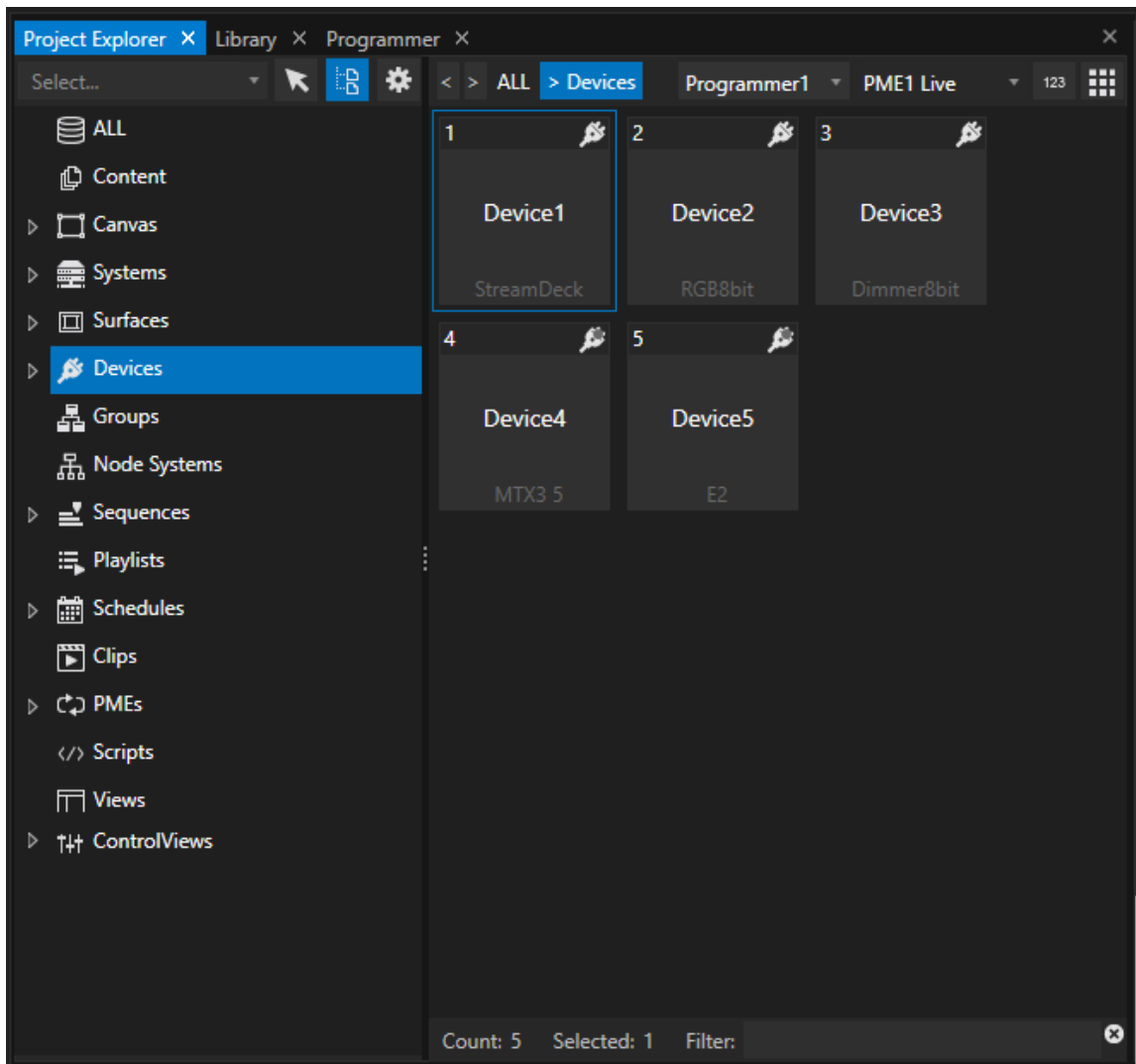
Devices that are shipped by ioversal are encrypted `.vxdx` format

**Directory and Folder:**

`C:\Users\Public\Documents\ioversal\Vertex\[Vertex Assembly Version]\Devices`

You are allowed to create own subfolders or copy your device into an already existing subfolder

## Devices in Project Explorer



The **device section in the project explorer** contains all devices that you have added to your project. Depending on the device type, a **connection status** gives you information about the connectivity.

**Status:**

**Red:** connection refused or not possible

**Orange:** try to establish connection

**Green:** connected

**Connection settings** like ports or IP Address could be done in the **inspector**.



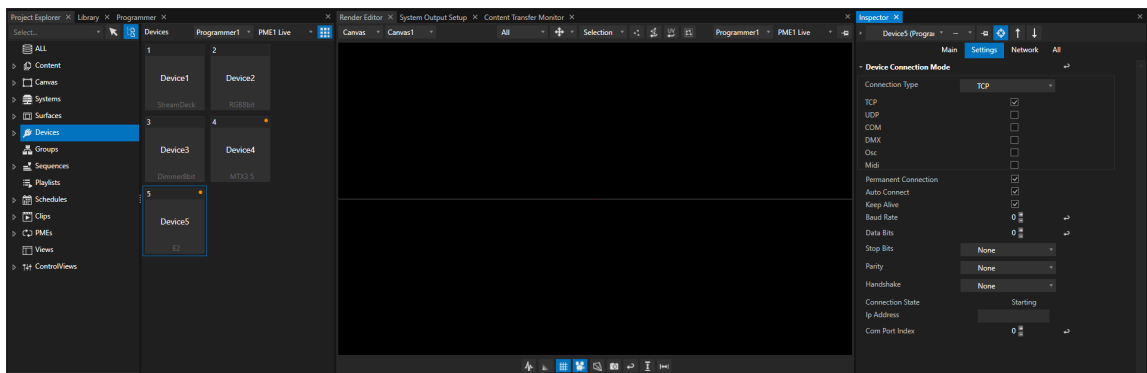


### Connection status into Project Explorer only for certain types of Devices

Some device types like DMX Devices or TCP Senders do not have a connection status by device design

For this Devices the status is not shown into Project Explorer

## Connectivity and Protocol Settings

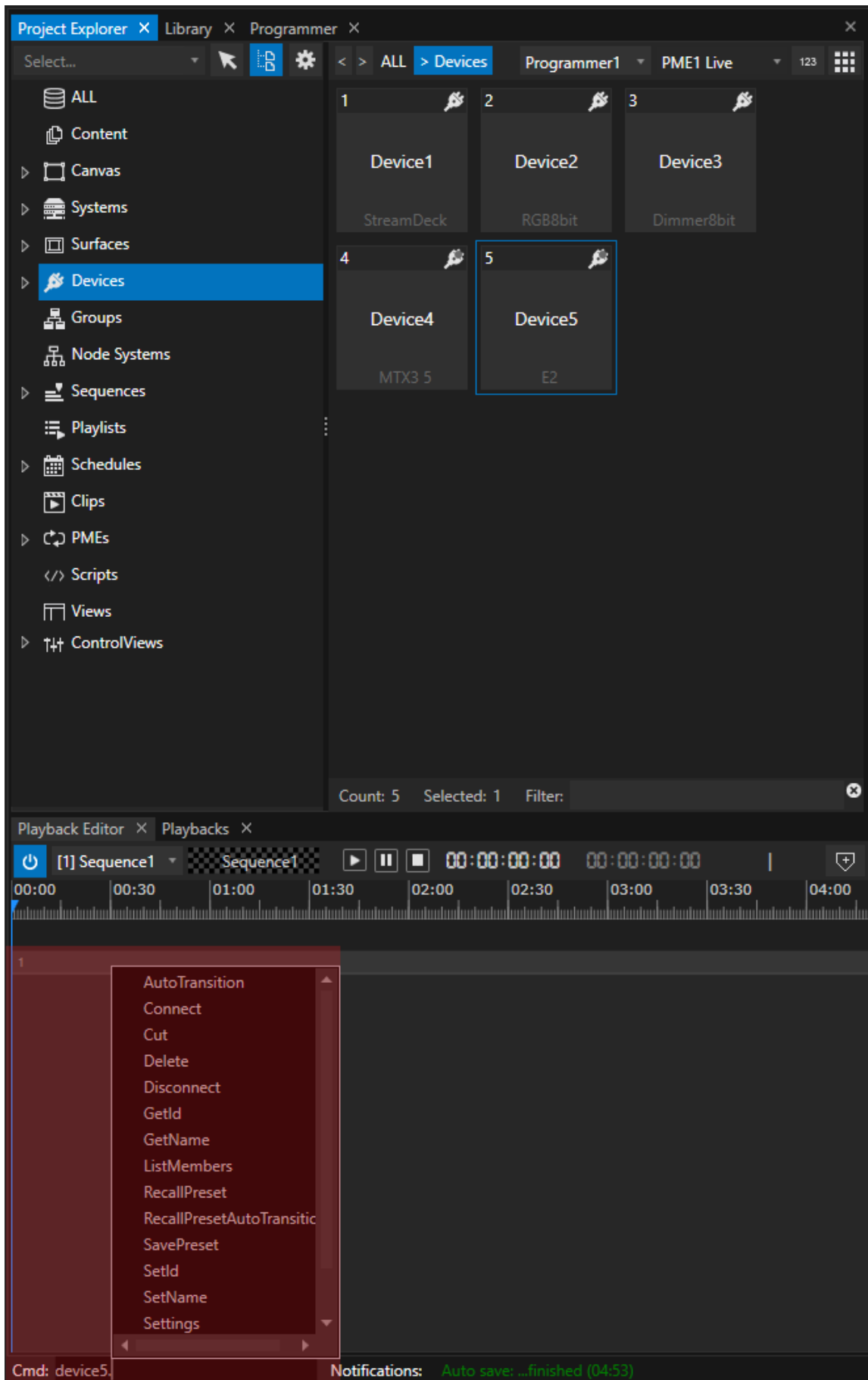


Select a Device in the project explorer and do the **connectivity and protocol settings** in the inspector.

Within the device library that is shipped with VERTEX, ioversal has tested the connectivity for most of the included devices. **All connectivity and protocol settings are set according to the manufacturer's specification.**

If you have problems to connect one of the devices, don't hesitate to contact the ioversal support ([support@ioversal.com](mailto:support@ioversal.com)) or open a ticket directly from within VERTEX (Main Menu -> help -> Create new Ticket).

## Use Devices with Script Commands



*Each device from a manufacturer has its own parameters and options.*

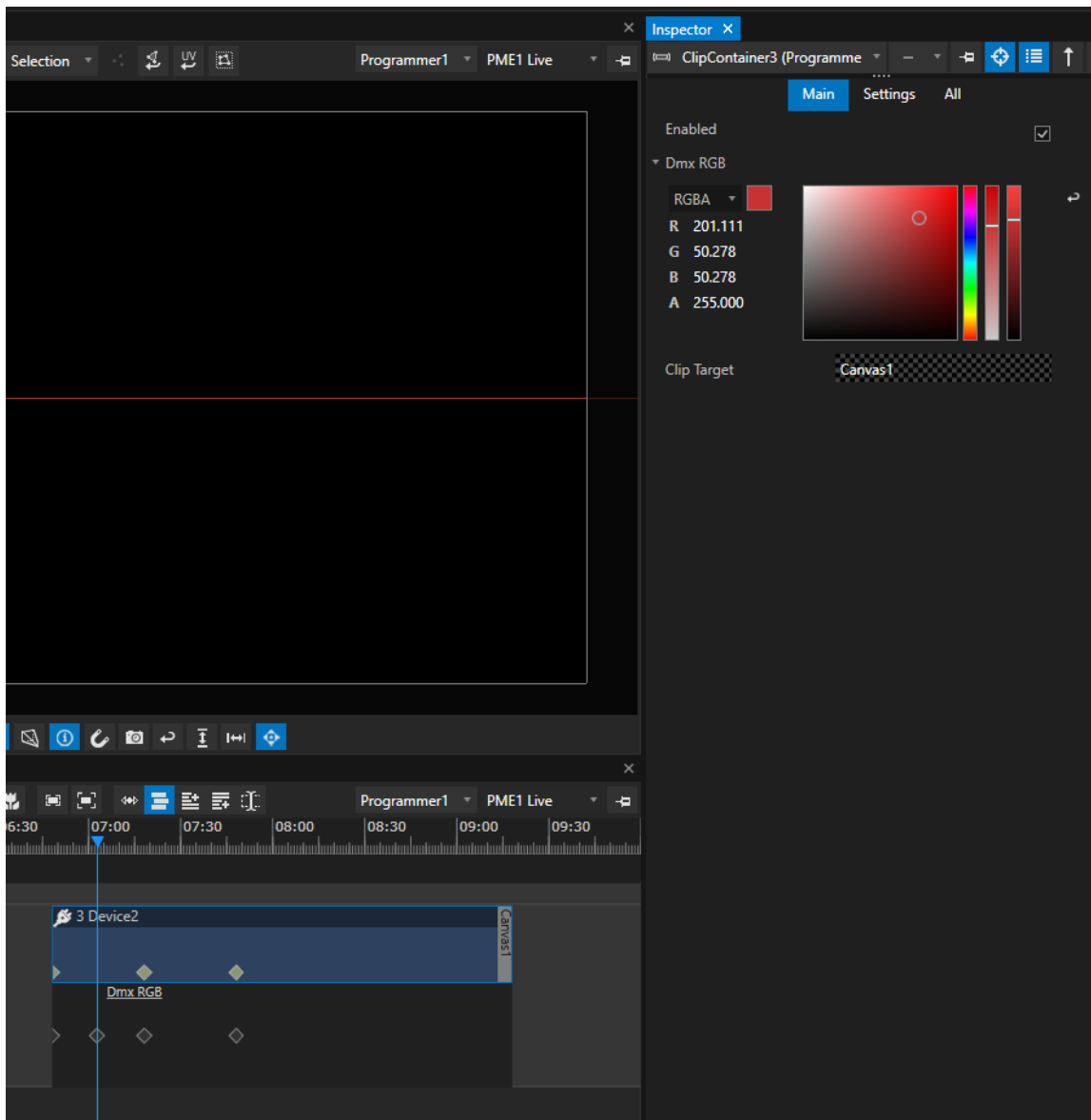
*This **options and parameters** are **accessible with VERTEX Script commands**.*

*Use script commands in cues, clip containers, on system startup, in a playlist, or in a trigger editor to script your device interaction.*

***For some kind of devices** - like DMX Devices also **keyframe animation** is possible.*

*For **some devices** (like e.g. the Elgato Stream Deck) **you only have to do the global settings into Inspector to work with.***

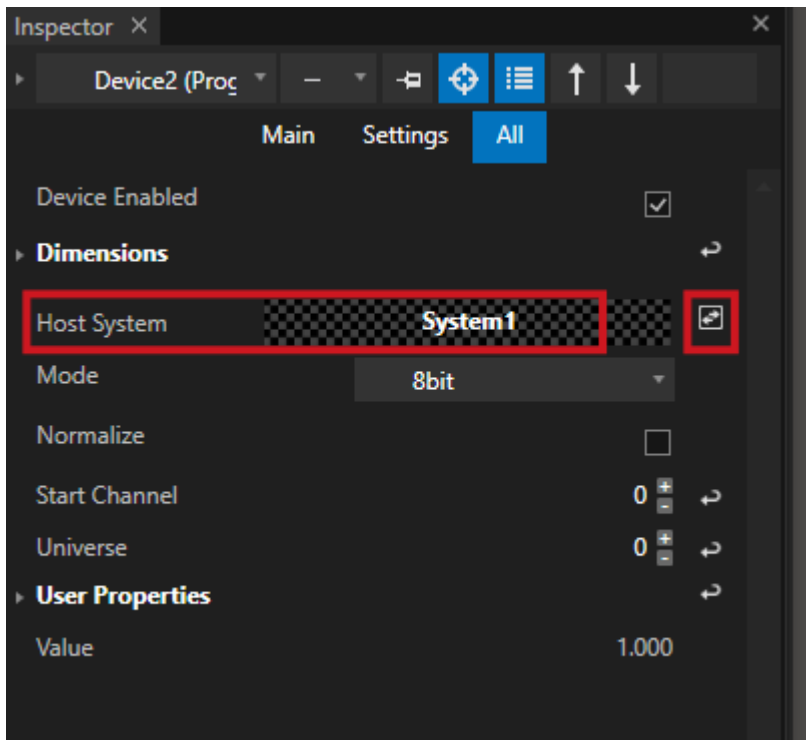
## Use Device as a Clip Container



Some Devices like **DMX Dimmer** and **DMR RGB** devices could be **animated with keyframes**.  
Just **drag the device from project explorer with your mouse and drop it into the playback editor**.  
A **clip container** is created.

## Connection Problems - Host System and Reconnect

- If there are **connection problems with your device**, please try a **reconnect** first of all.



- Select your device in the *inspector*
- Every Device has a **property for Host System** (The VERTEX system your device is connected to). Please first check if this is correctly set  
If not and if you are working with **multiple systems**, drag another system from project explorer to this property field.
- Next to this property, there is a **reconnect button** that refreshes the connection to this device.

## Reconnect by Script

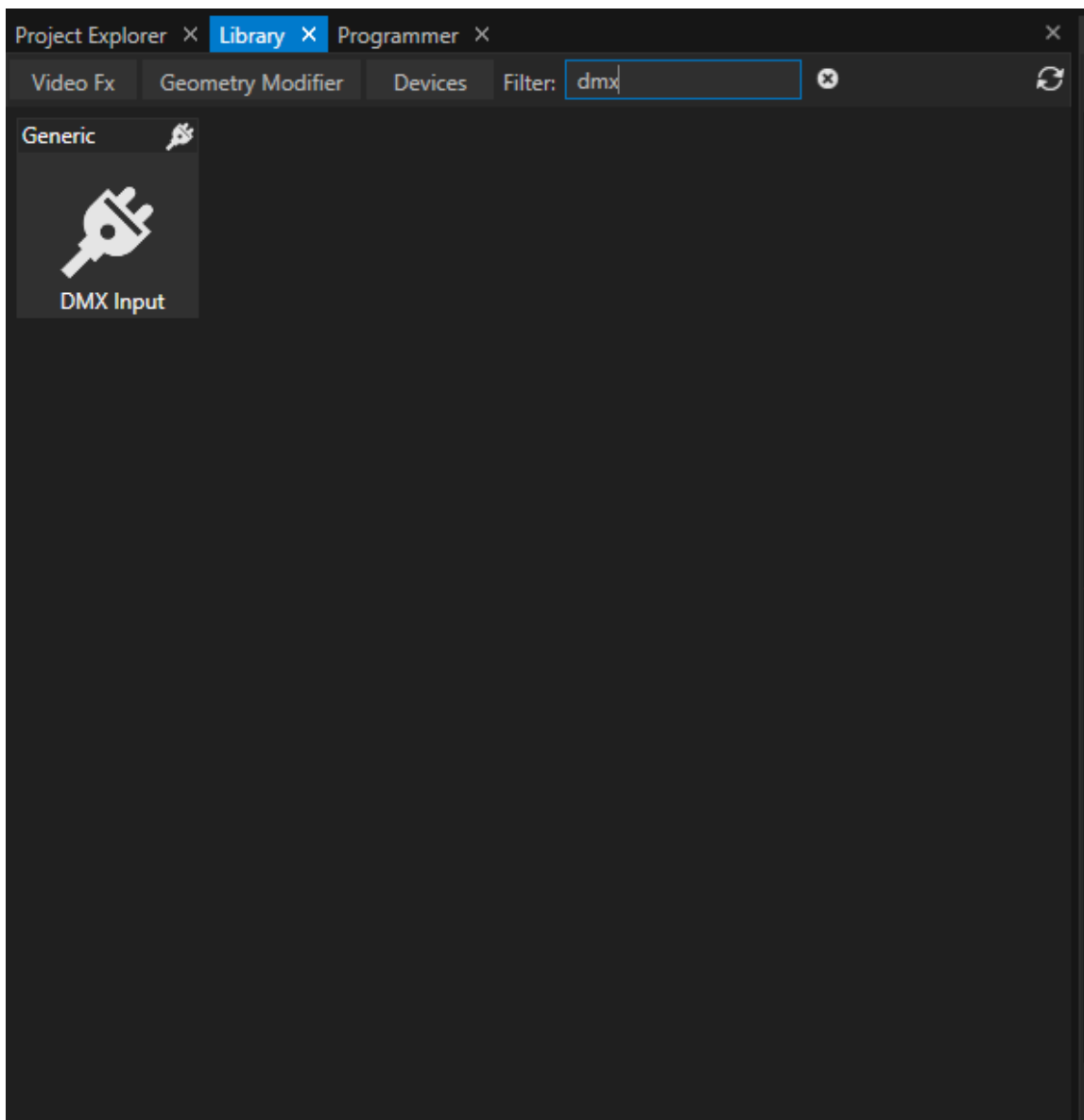
- A reconnect is also possible by script with the command **ResetConnection**

```
Device1.ResetConnection
```

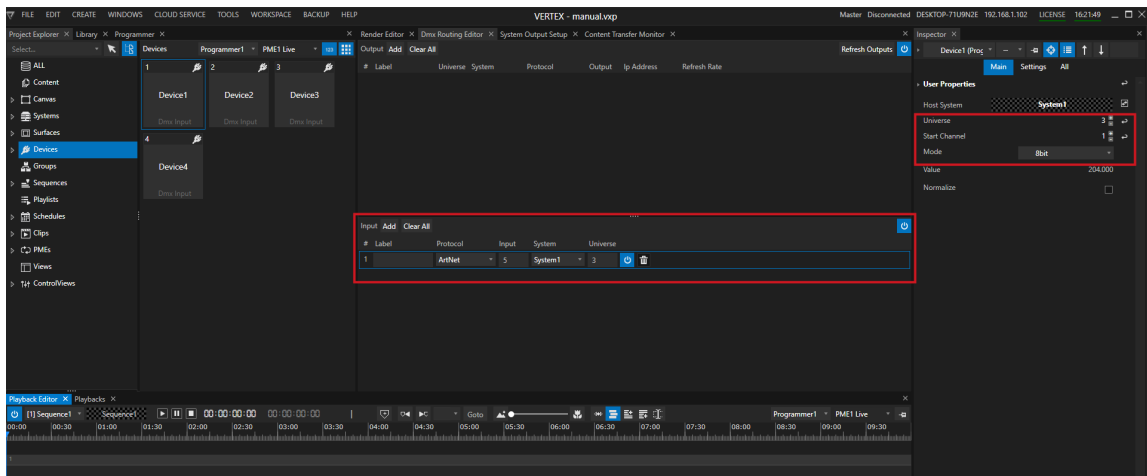
### 5.3.13.1 DMX Input Device

- Use DMX Input Devices to **trigger, wire or script interactions based on values of a DMX Channel**
- You are able to **listen on DMX Values** of a predefined DMX Start Address and Universe
- Use the **incoming DMX Values to influence other properties into VERTEX** or to trigger an action

## Add Device and do Settings



- Go to the Library Editor
- Select the tab "Devices"
- Search for "DMX Input"
- Right-Click on the Device and add one or multiple to your project



Example of a DMX Device Setting: There is DMX Input Routing. VERTEX is listening on Art-Net Universe 5. Data from this Universe is internally routed to VERTEX Universe 3. DMX Input Device 1 is set to VERTEX Universe 3, Start Channel 1.

- Go to the Device section in the [Project Explorer](#)
- Select on or multiple of your DMX Input Devices
- Set VERTEX universe number and start channel and define the DMX mode (Default 8 Bit, 16 Bit, 32 Bit...)



### Normalize Input

Enable "Normalize" in the Device settings, if you want to control or wire a property into VERTEX that has a value range from 0 to 1 (e.g. opacity).

All incoming DMX Data is normalized into the range from 0 to 1.

- Ensure, that a valid [DMX Input Routing](#) exists.

## Working with DMX Input Devices

As for every Device, there are different options to work with the incoming Data.

Here some Examples:

### 1. Scripting

The Opacity of Clip Container 1 is controlled by the incoming DMX Data from Device 1

The opacity value range is 0 to 1. The "Normalize" data option is enabled for the input Device

A [Script Command](#) for this scenario could be like this:

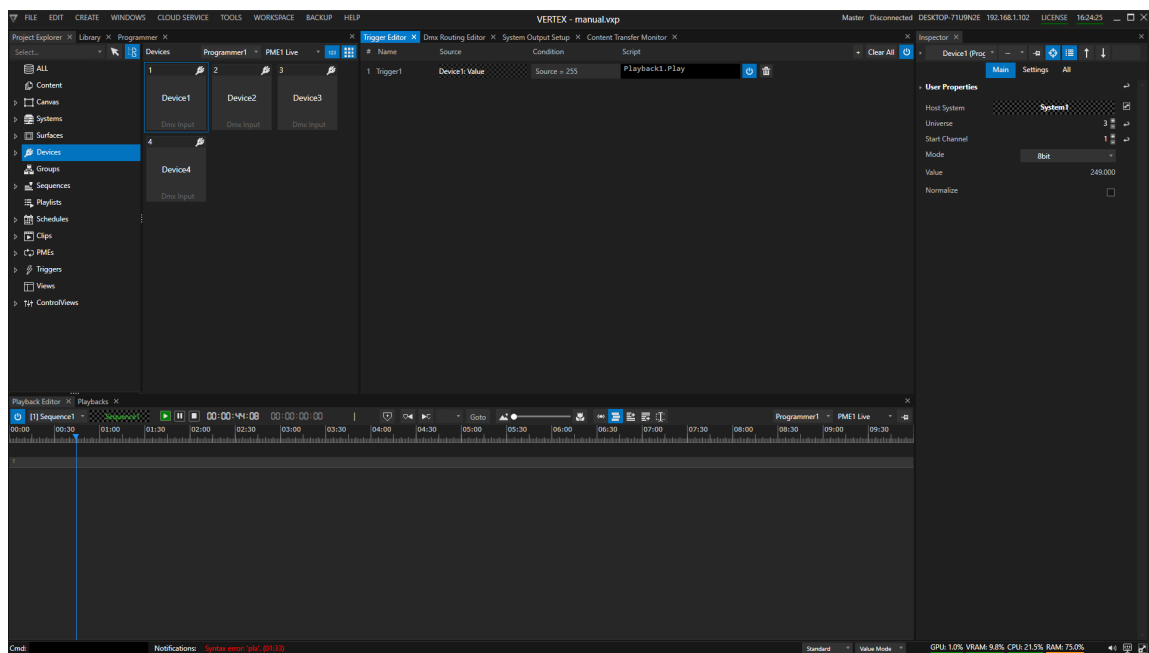
```
Sequence1.ClipContainer1.Opacity.Value = Device1.Settings.Value.Value
```

### 2. Trigger

With help of the [Trigger Editor](#) you can easily build Triggers. An defined action is executed if a condition for a source is reached

Of course this also works with DMX Input Devices

- Open the [Trigger Editor](#)
- Drag the e.g. Value property from Inspector to the Source field
- Enter a [Condition](#)
- Enter a [Script](#) that should be executed



Example for the usage of a DMX Input Device:  
If DMX-Value of Universe 3/Startchannel 1 is 255, Playback 1 is started

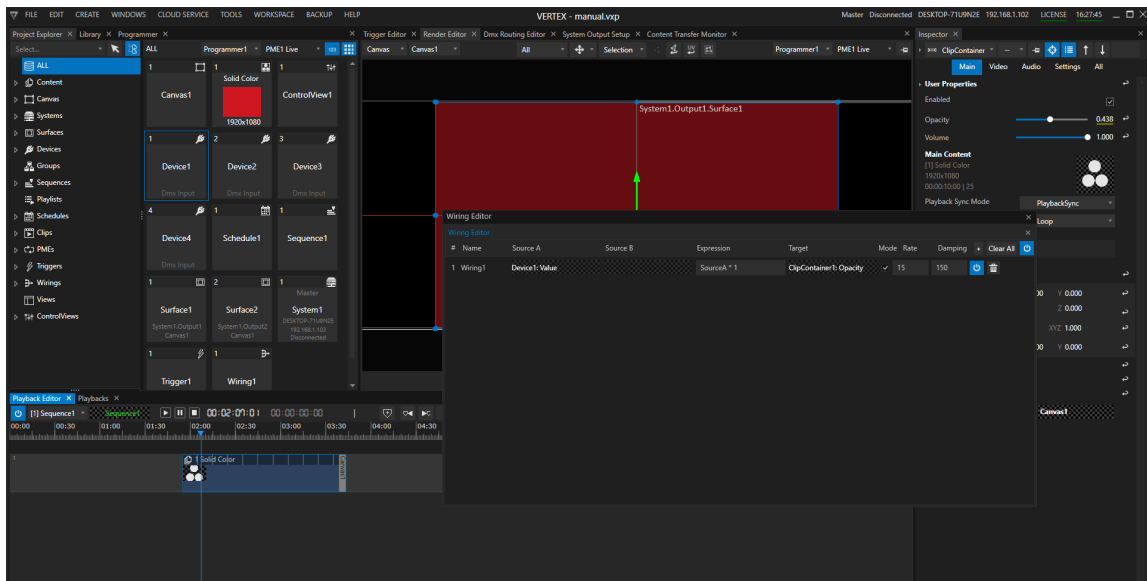
### 3. Wiring

[Wiring](#) gives you the option to connect Values of different properties.

Of course this also works for DMX-Input Devices

- Open the [Wiring Editor](#)
- Drag a DMX Device Property from Inspector to e.g. the Source field
- Drag another property into target field
- Add an expression





Example of a Wiring

The Value of the DMX Input Device 1 ( Normalized) is wired to the Opacity of Clip Container 1.  
The Opacity now is controlled by a DMX Channel.

### 5.3.13.2 DMX Output Devices

- With **different types of DMX Output Devices** you are able to send out DMX Data
- Integrate DMX Devices to your show, **program lighting scenarios based on keyframes** or work with them in **programmer mode**
- Create a **group of Devices** and control them together with only one Clip Container

## Workflow

### Add from Library

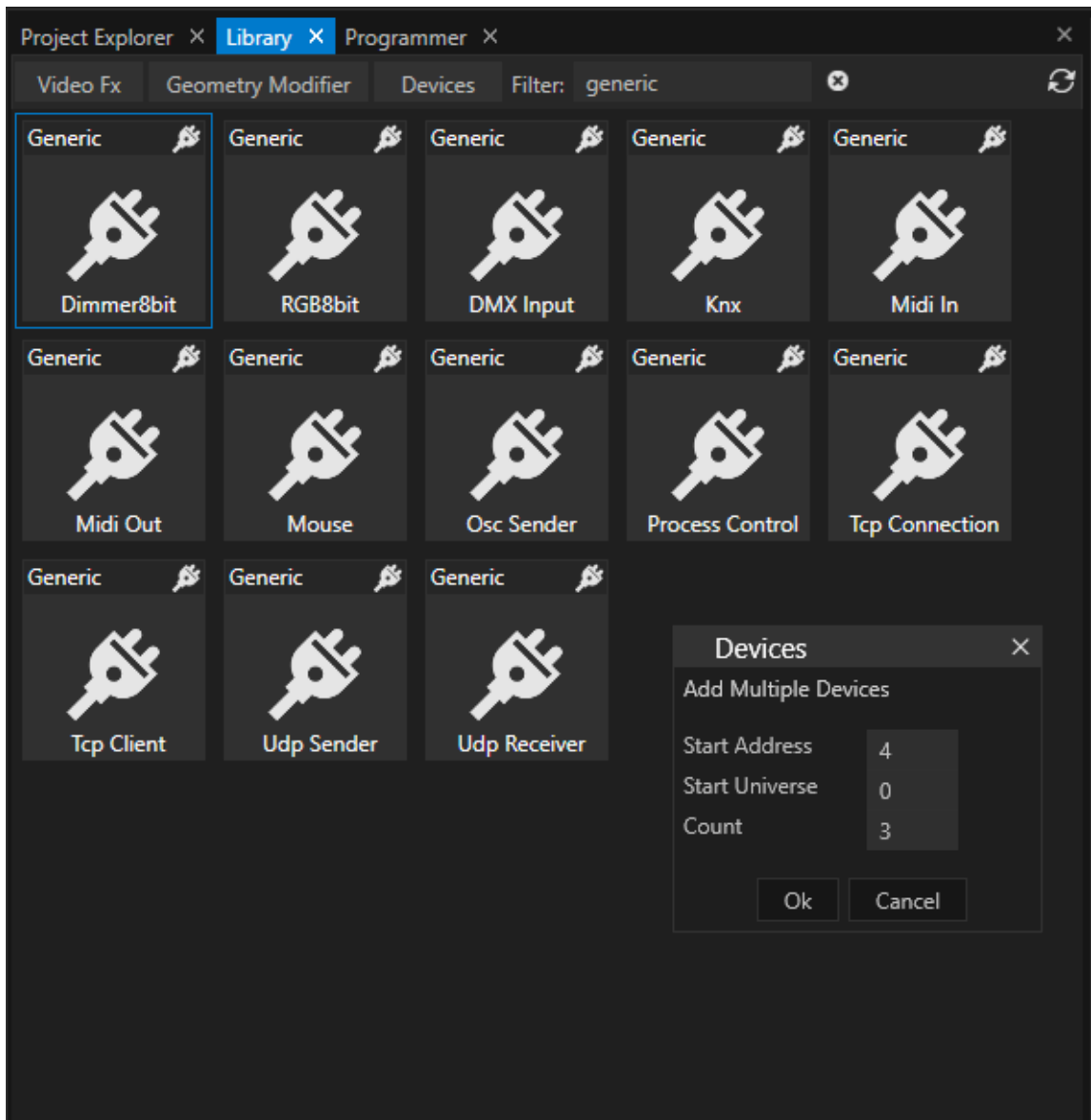
- Open the **Library Editor**
- Select the **Devices** Filter there
- **Search for your DMX Device**



#### **DMX Devices and VERTEX Library**

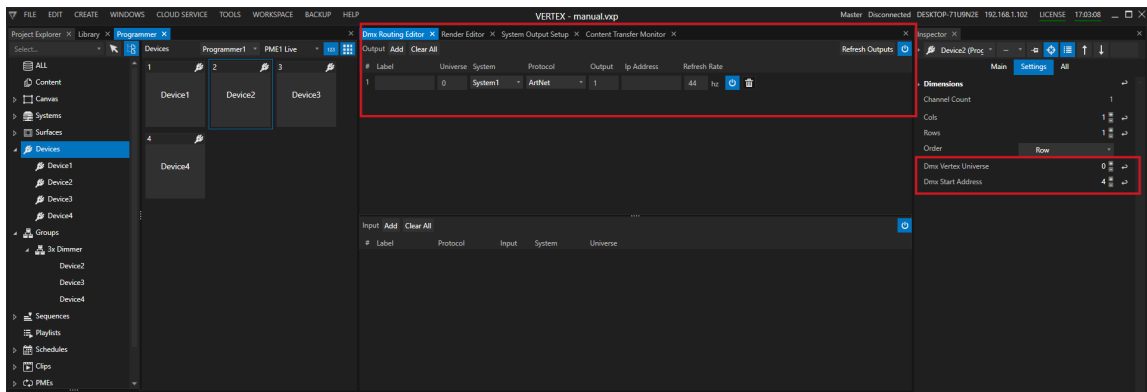
Depending on the VERTEX Assembly version, the available number of different DMX Devices could change. Basically, VERTEX is shipped with Generic standard devices: a 8 Bit Dimmer and and 8 Bit RGB Device. If you require other DMX Devices for your project, just write us an email to [support@ioversal.com](mailto:support@ioversal.com) with type and in the most ideal case directly with the DMX table :-)

- **Right-Click** on the Device you want to add
- Select "**add to project**" or "**add multiple**"
- If you only have **added one device**: select the device into Inspector to set DMX Universe and Start address
- If you have select "**added Multiple**", a dialog window opens. Just add the **start address and universe** there and the number of devices you want to add.  
The first device will start with start address and universe you have set. For all other devices, start address and universe are automatically set by VERTEX.



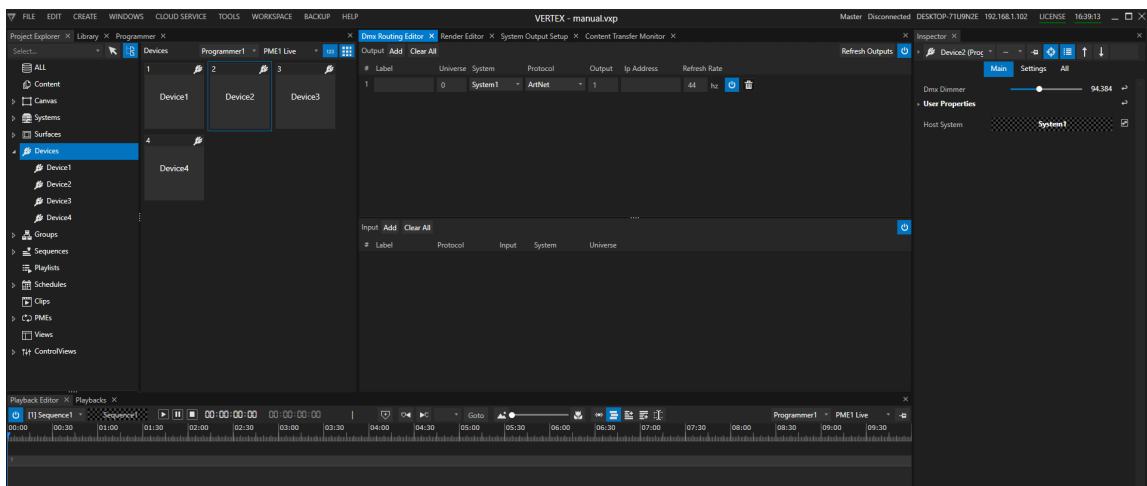
## Check DMX Routing and Start Addresses

- Remember that a DMX output only will work, when a **valid [DMX routing](#)** is created. If this was not done before, please create a routing now.  
Into the [DMX Routing Editor](#) you define, which protocol is used to send DMX Data out. To send physical DMX-512, ioversal provides an USB-DMX interface, the [<%DMX\\_IO%>](#)
- Go to the Device section of the Project Explorer. select a DMX device there and set or check the **DMX Start Address** or the **Start Universe** in the Inspector.

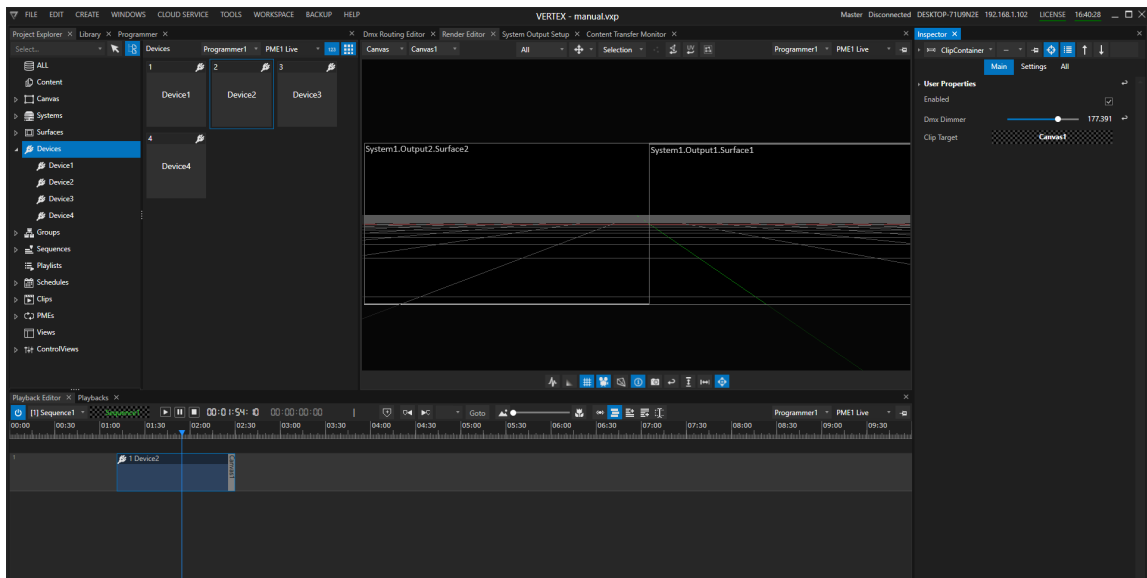


## Send Out Global Values

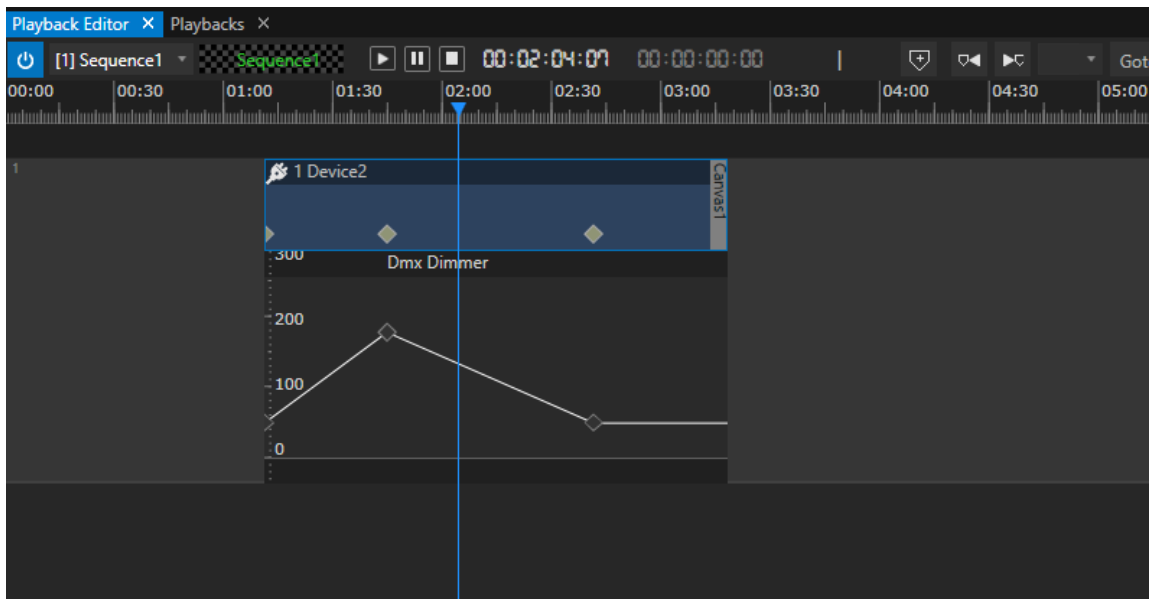
- You are able to send out a global value for your DMX device, e.g. for testing or if there generally should be a minimal value for such devices all the time.
- To set values globally, select a device directly into Project Explorer and set the DMX Value in the Inspector.
- This value is sent by vertex permanently



## Work with Clip Containers and Keyframes



- A combination of [Clip Containers](#) with [Keyframes](#) allows you to create more complex lighting scenarios, color transitions or an integration of lighting into your video show.
- Device values from a Clip Container temporarily **overwrites** [global values](#) (if such are set)
- Just drag a device from Project Explorer into the playback Editor to **create a new Clip Container**
- Double click on the Clip Container to open the **Keyframe Editor** (to learn more about Keyframe animation, please read the topic [Keyframes](#))



- If you prefer a **workflow from a Lighting Desk** - you can also work in **Programmer Mode**:  
Change all values and create your scene. All changes are temporarily stored as a list in the **Programmer**. When your scene is final, you can save the programmer list as a scene: VERTEX automatically creates the Keyframes for all changed values at the Playhead position.

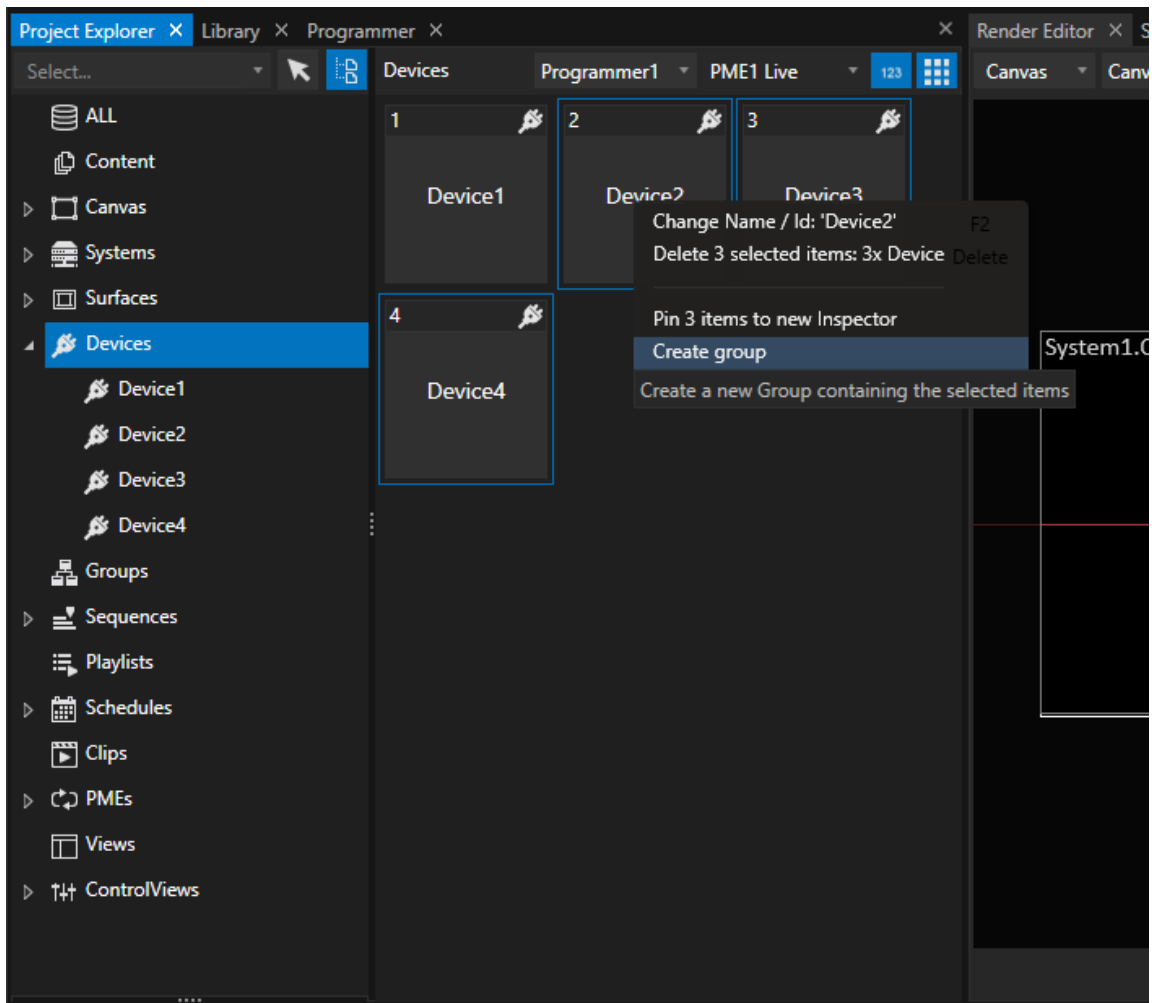
## Groups

- Manage and summarize your DMX Output Devices in Groups. Make value changes for all devices of a group

### Create a Group

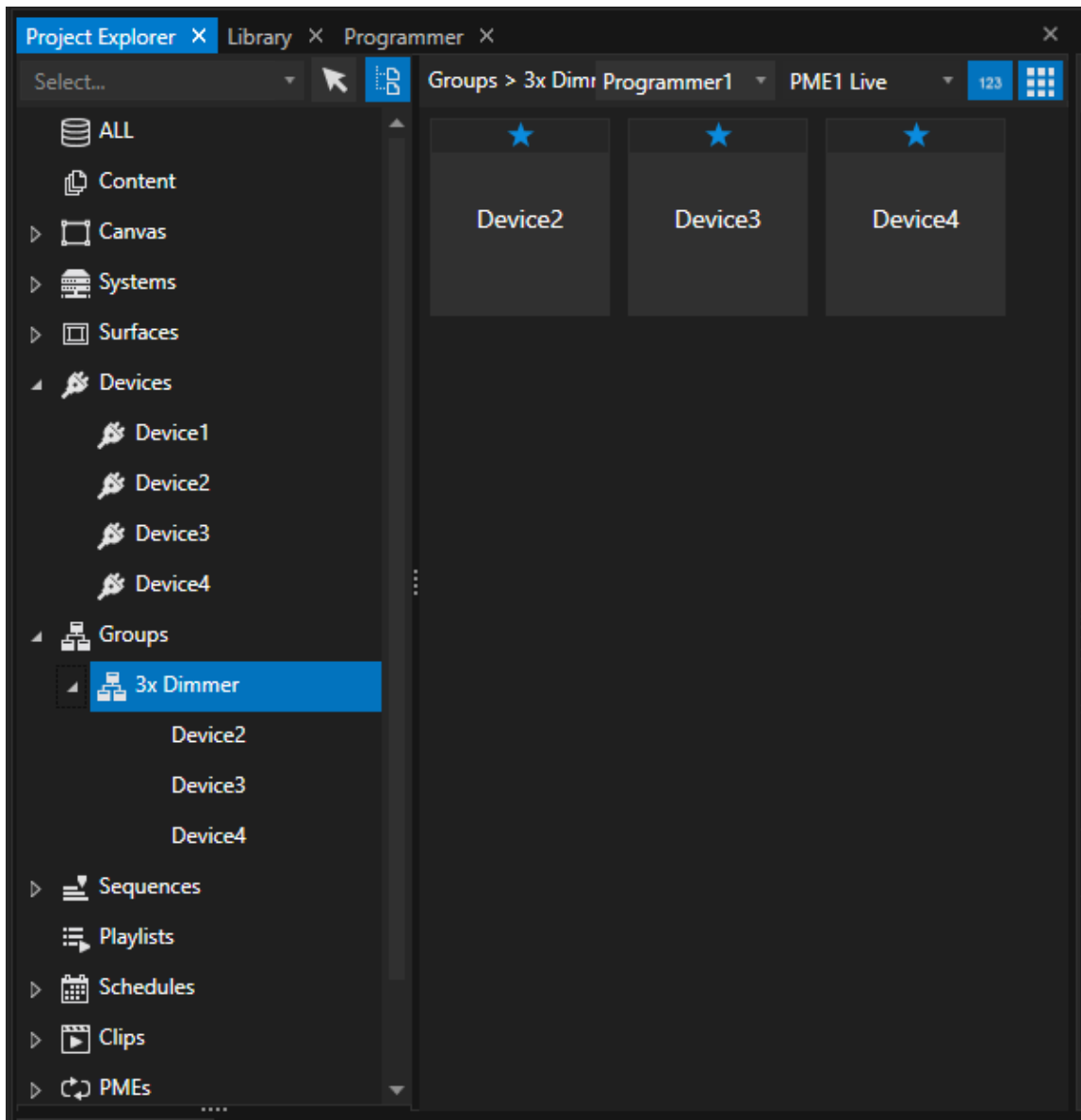
**There are 2 ways to create a group and assign Devices to it:**

- Go to the Device section into the Project Explorer
- Mark all devices that should be member of a group
- Right-Click and select "Create Group"
- A new Group that contains the selected Devices is created. You can find, select or rename the group in the "Group" section of the Project Explorer



or:

- Go to the "Groups" section in the Project Explorer
- Right-Click and open the context menu
- Choose "Add new" there
- Drag devices from Device section of the Project Explorer to this group



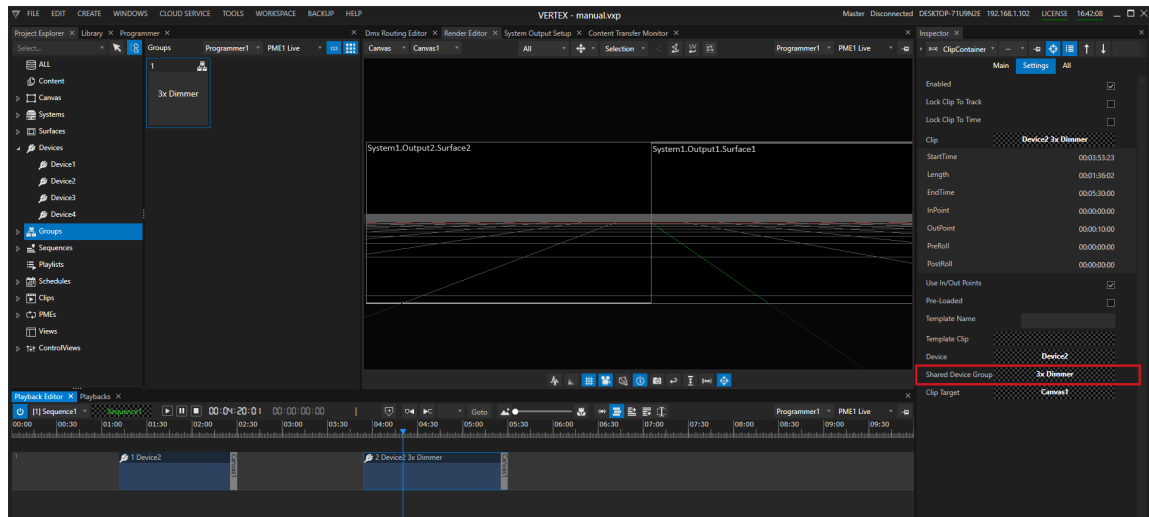
## Work with a Group of DMX-Devices

*When working with a group, you are able to set DMX Values for all group members with only one Clip Container.*

- Drag a Group from Project Explorer to the Playback Editor
- All changes you do for this Clip Container are adopted for all devices of this Group
- If values apply to a group, you can identify this with the "SharedDevice Group" Property. If there is a group set, all values of the clip container are also assigned to the group members.



- You can also transfer the settings for a DMX Device to a Group afterward by dragging a Group from Project Explorer to the "SharedDevice Group" field of an already existing Clip Container (that already hosts a DMX Device)



### 5.3.13.3 Powerpoint Remote

- VERTEX is able to **control a Microsoft Powerpoint application remotely** over network
- A **background service** is able to start Microsoft Powerpoint, load a .pptx-file, jump to a slide or set a presentation into full screen
- The background service can be controlled by a **PPT Remote Device** that is part of the **VERTEX device library**

## Use Cases

- You need to integrate Powerpoint with its full functionality into your project and not even import and read Powerpoint files
- Use VERTEX as master control system to remote start and control a PC with Powerpoint in your installation
- Capture the output of a Powerpoint PC as Live Input into VERTEX and control slide changes from your VERTEX timeline
- Read out notes of a Powerpoint file and display this notes as text on another screen - e.g. backstage

## What you need

- a PC with Microsoft Powerpoint installed and Microsoft Windows as operating system
- a PC with VERTEX installed
- a network connection between both PCs (Wireless LAN or cabled LAN)

**VERTEX and Powerpoint on the same PC**

For Tests or even smaller projects with only one PC, you are even able to remote control a Microsoft Powerpoint application on the same PC on which VERTEX is running. Just use your local IP address as IP address of the PPT Remote Device

## Setup

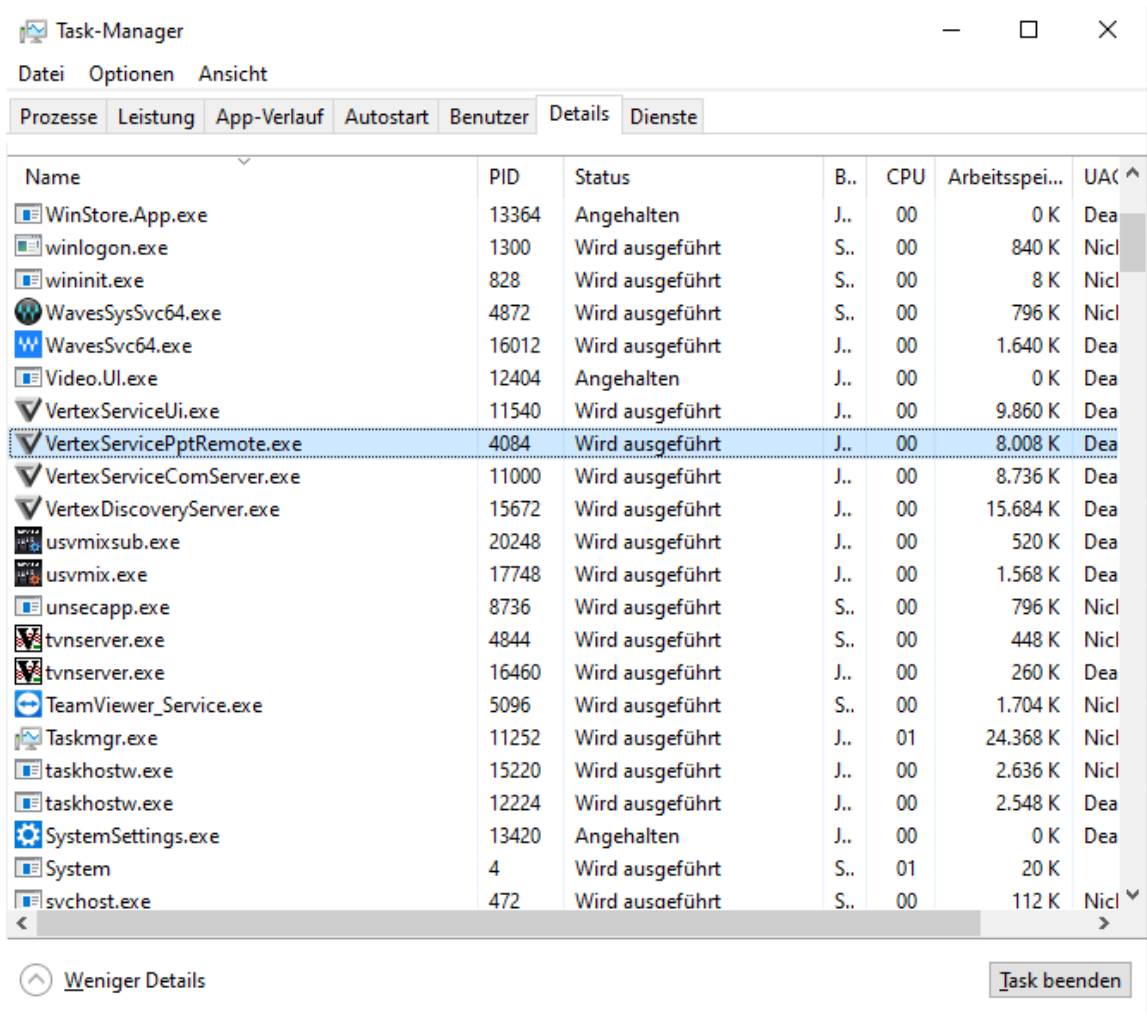
### 1. Prepare Powerpoint PC

- **Install Vertex on the PC where Microsoft Powerpoint already is installed.**
- **Check your Network settings**, prefer a static IP address to avoid address changes after a reboot

**VERTEX Trial Version is enough**

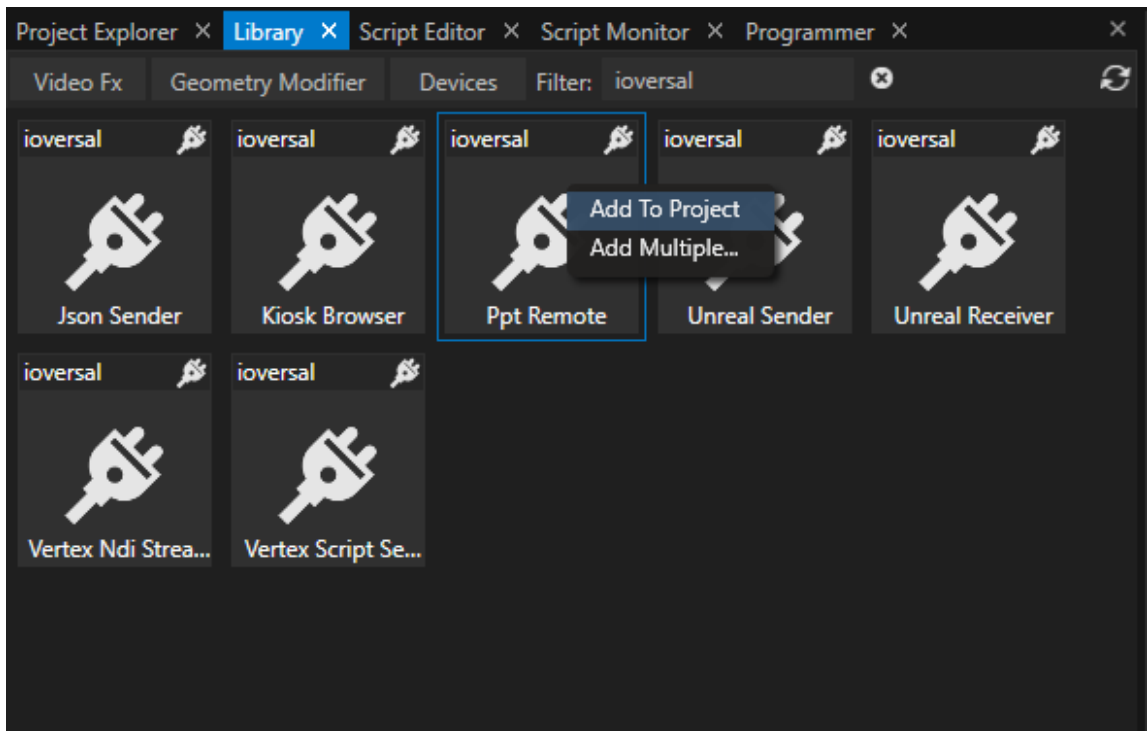
You don't need any Vertex license on the Powerpoint PC. The Vertex Trial version is enough. The Vertex installer will install a Windows background service to remote control Microsoft Powerpoint on this PC.

- **Reboot** the PC after installation
- Open Windows Task Manager and Check if VertexServicePptRemote.exe is running into background
- You don't have to start the main Vertex application on this computer. To remote control Powerpoint it is only necessary that all VERTEX background services are running.

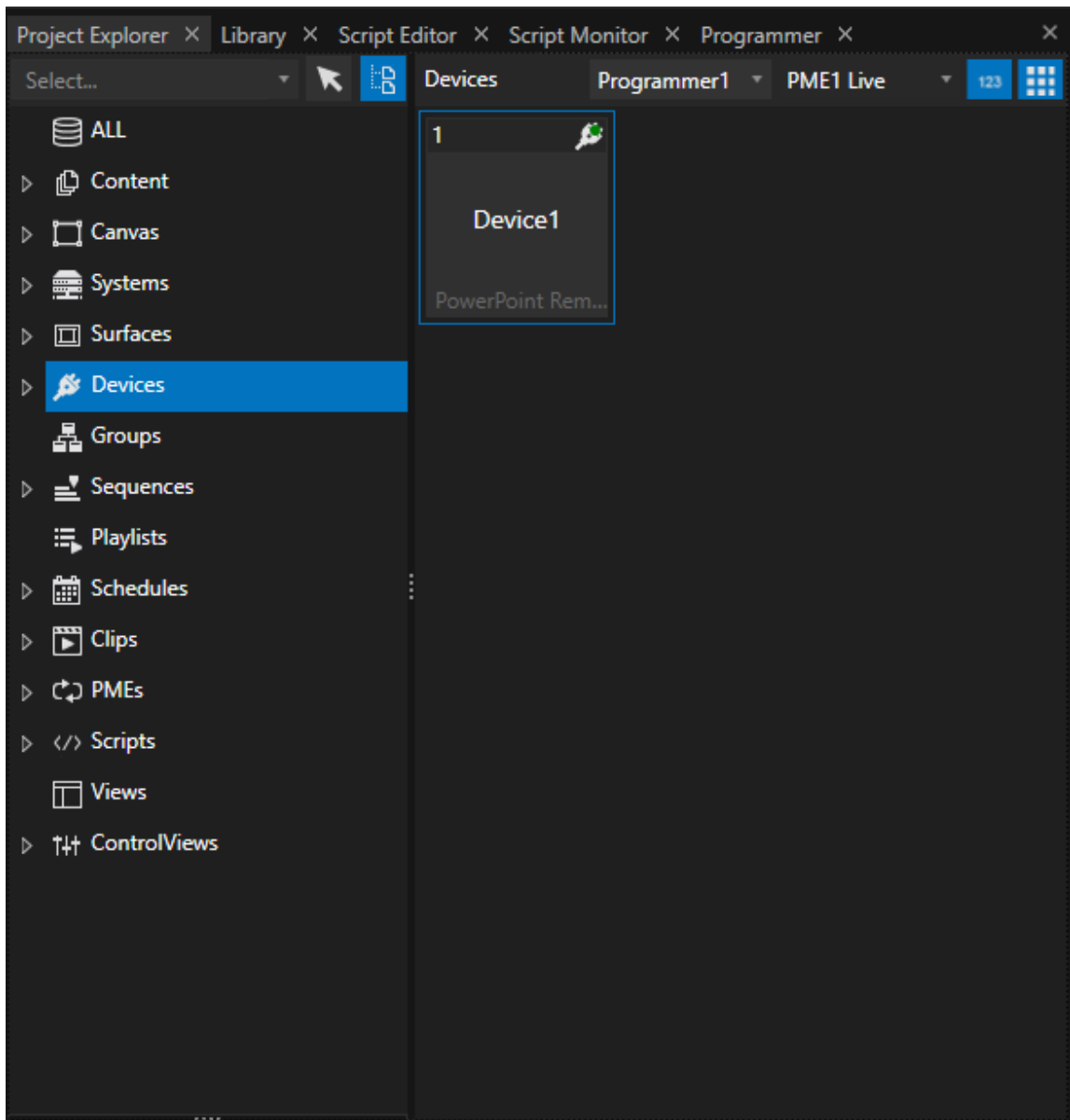


## 2. Prepare your VERTEX Project

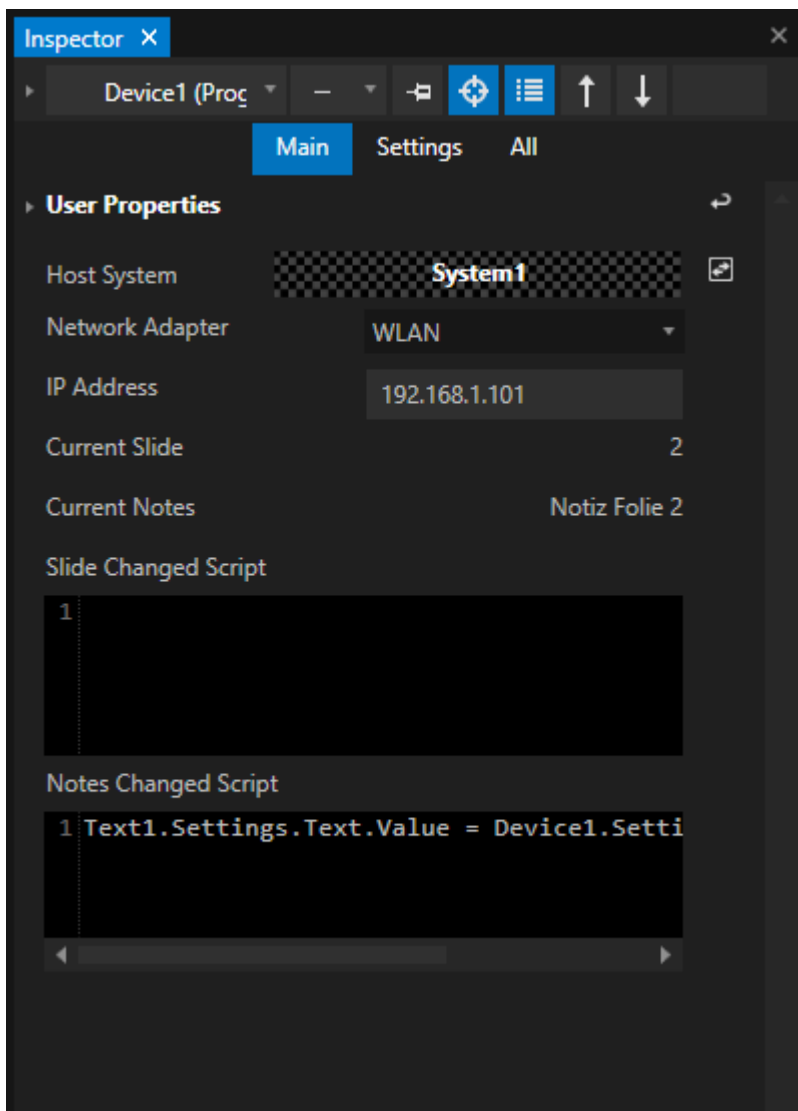
- Search for the **"Ppt Remote"** device into the [Library Editor](#)
- **Open the context** menu by a right click
- **Add** this device to your project



- Select the device into Project Explorer



- First configure the **Network Adapter of the Ppt Remote Device** in the **Inspector**
- Use the reconnect button and test the connection



- Now the Device should be ready to work with

## Examples

*Launch a specific Powerpoint file and go into presentation mode*

```
Device1.Launch C:\Users\User1\Desktop\test.pptx
```

*Quit whole Microsoft Powerpoint application*

```
Device1.Quit
```

**End presentation mode of Powerpoint but keep Powerpoint application running**

```
Device1.EndPresentation
```

**Go to Slide 2**

```
Device1.GotoSlide 2
```

**Example for a more complex start script**

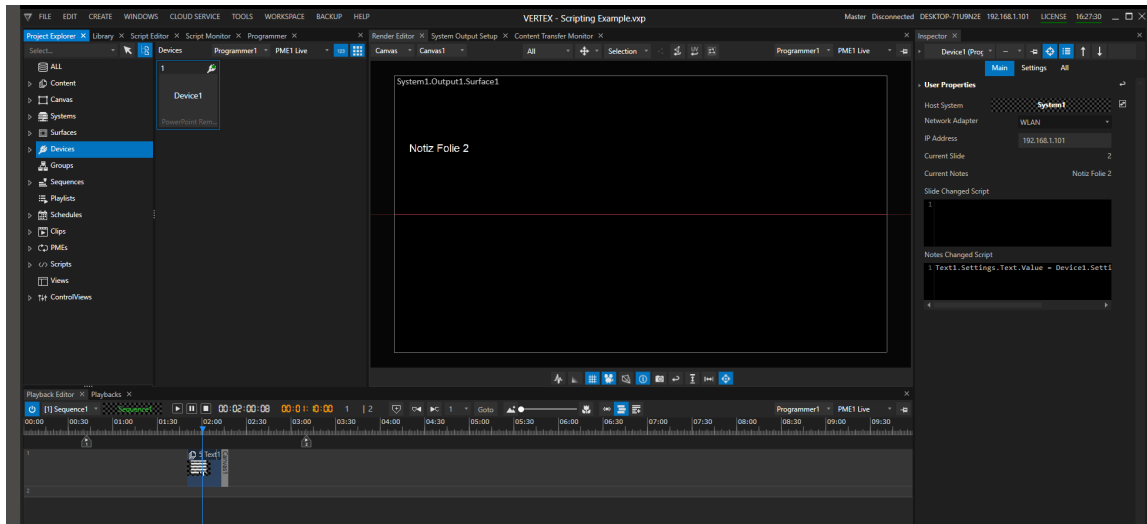
- Reset Connection (to be sure that connection is ready and working) and wait 1 second
- Set Volume of Powerpoint PC to 0 and wait 1 second
- Launch test.pptx file from Desktop of User1 and wait 3 seconds
- Go to next slide

```
Device1.ResetConnection  
wait 1  
Device1.SetVolume 0  
wait 1  
Device1.Launch C:\Users\User1\Desktop\test.pptx  
Wait 3  
Device1.NextSlide
```

**Show Notes from Powerpoint file as text content into VERTEX**

- The current powerpoint notes should be displayed as text content into VERTEX
- For our usecase, the text content into VERTEX should only be changed when the notes of the pptx are changed
- The Ppt Remote Device offers 2 options: "Slide Change Script" and "Notes Changed Script". For our usecase, add the following code snippet as Notes Changed Script

```
Content1.Settings.Text.Value = Device1.Settings.CurrentNotes.Value
```

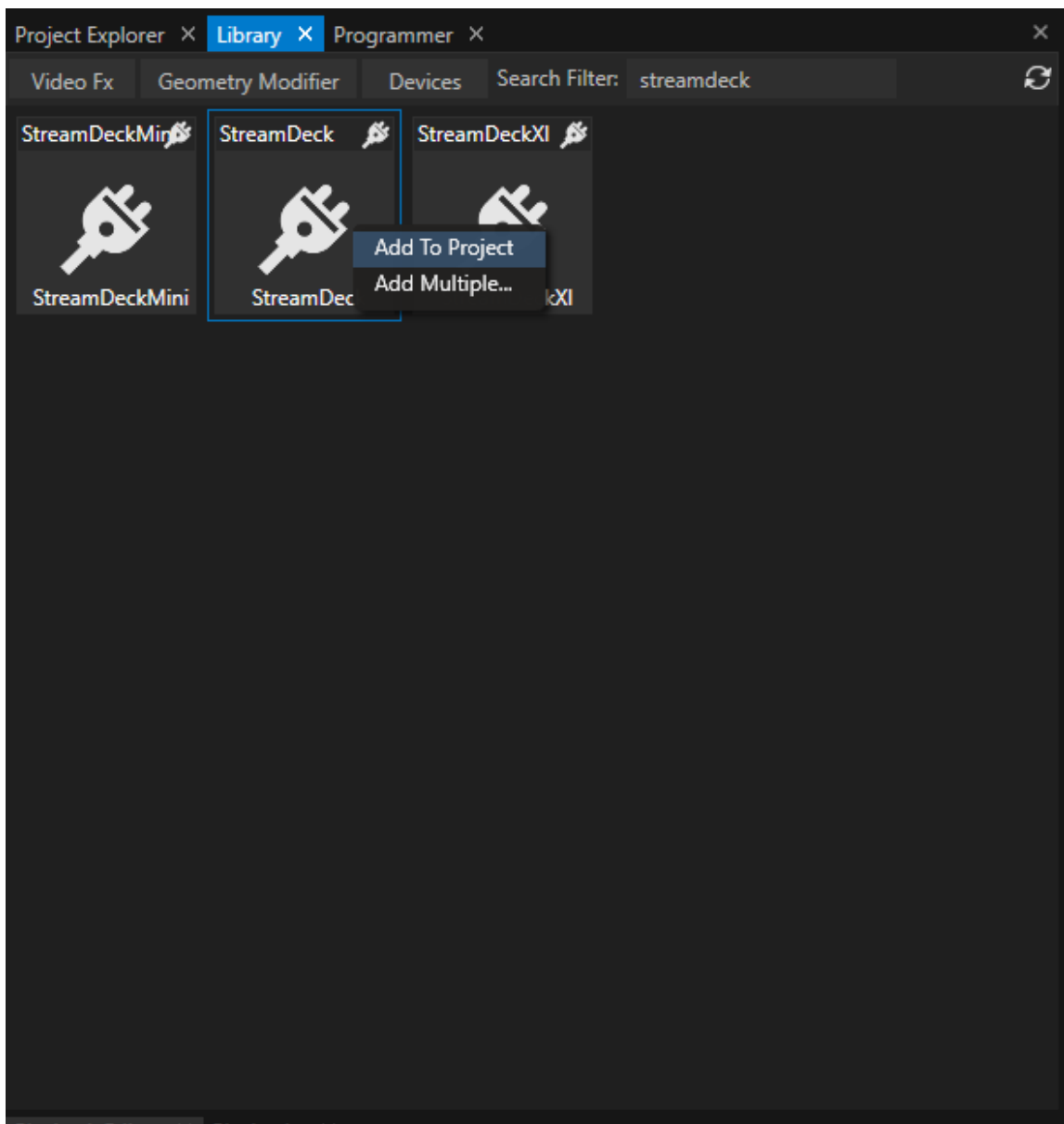


#### 5.3.13.4 Stream Deck

- VERTEX is shipped with **device Library elements** for the **Elagato Stream Deck, Stream Deck XL and Stream Deck mini**
- **No Driver installation** is required, just plug in the Stream Deck and start VERTEX
- **Settings for all Buttons are done in VERTEX - Script Commands** are used to trigger actions

## How to add a Stream Deck





1. **Connect the Stream Deck** to your PC#

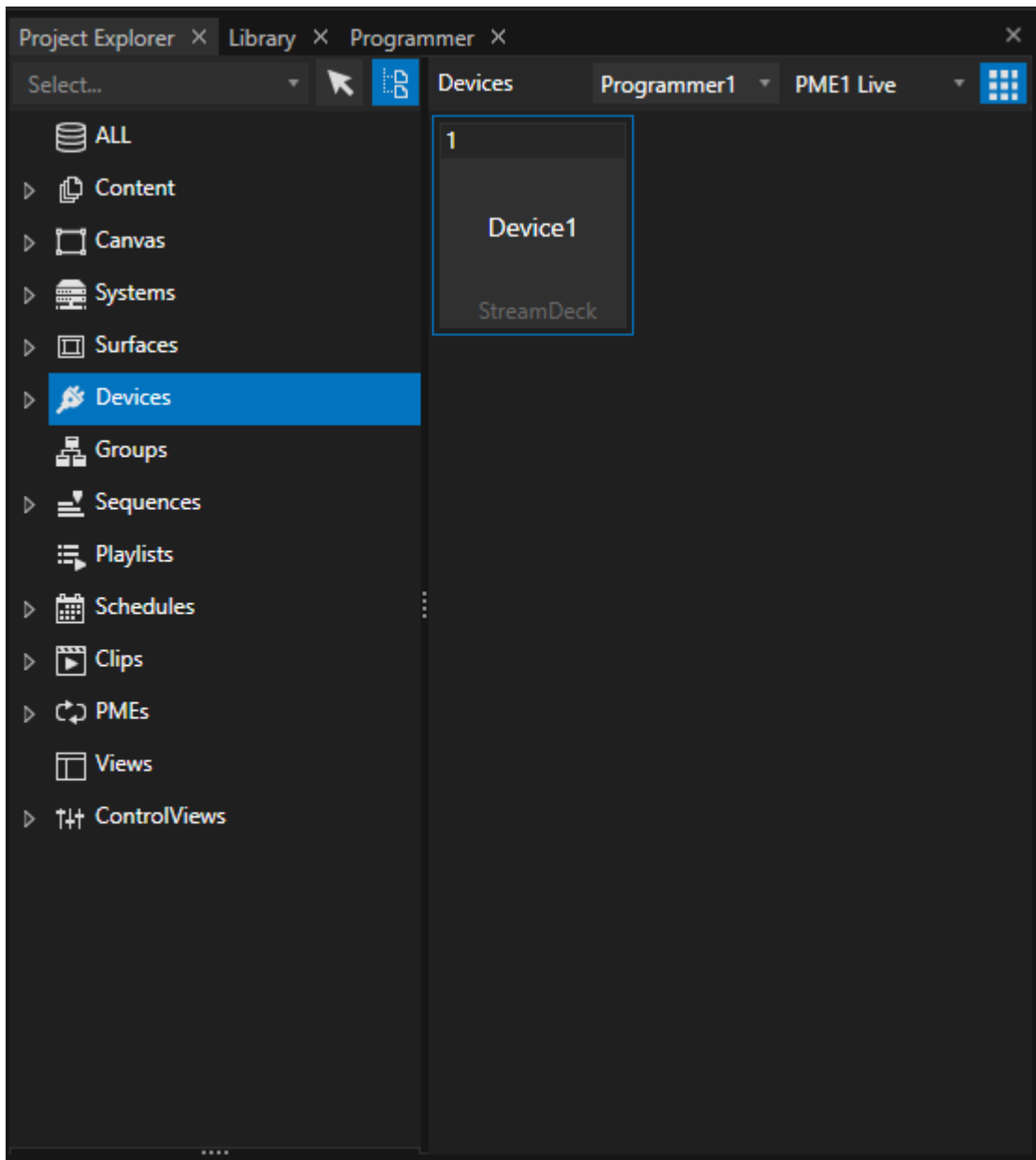


**Close Elgato "Stream Deck for Windows" Application and all other Applications that access your Stream Deck**

VERTEX directly accesses and talks to your Stream Deck device. Avoid side effects from other applications - like overwritten button designs - by closing all other applications before.!

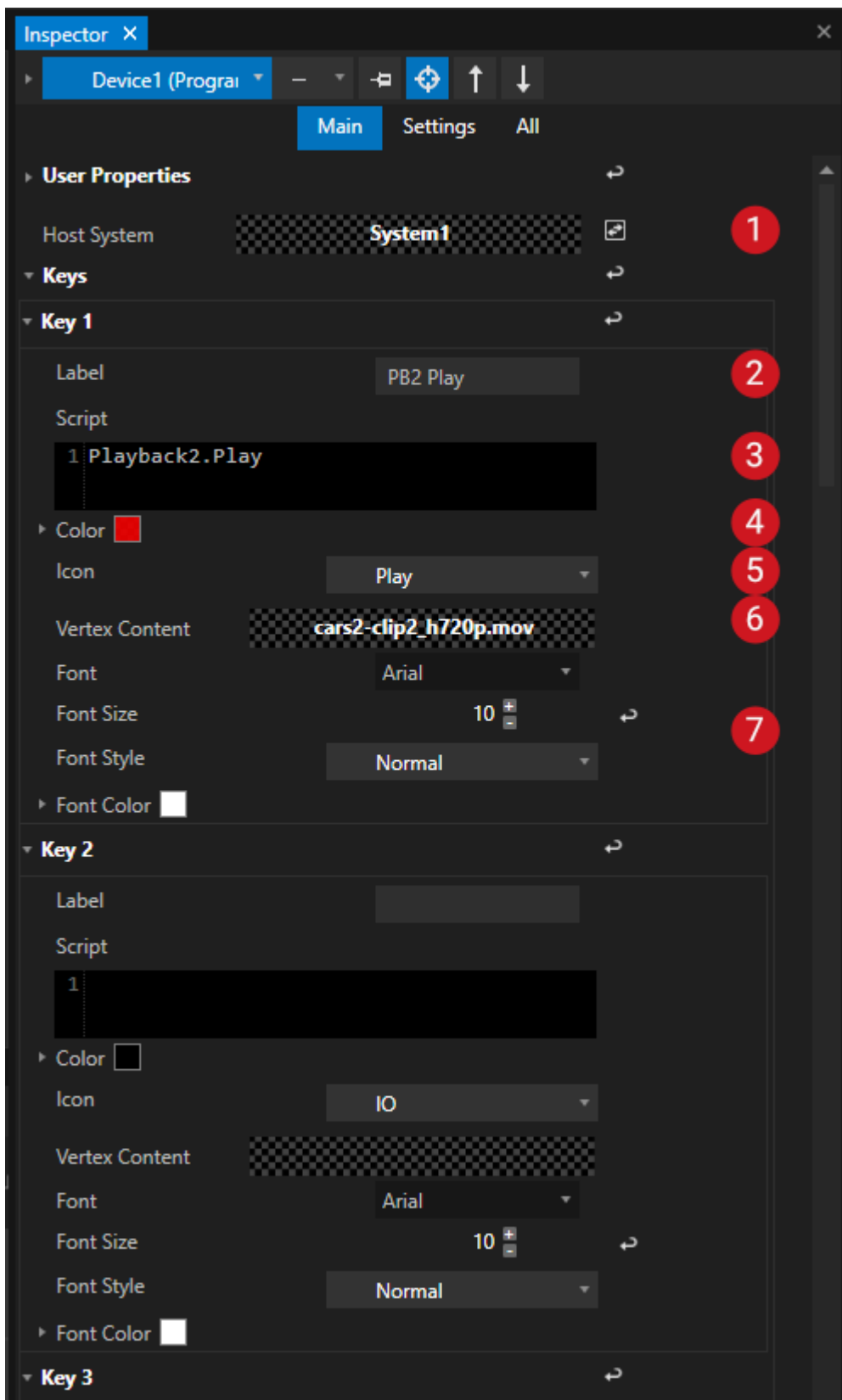
Especially the "Stream Deck for Windows" application from Elgato influences a Stream Deck that is used within VERTEX. Please also terminate this application before you start with your VERTEX project.

2. **Start VERTEX**
3. Open **Library Editor** and search for the **Stream Deck Device**
4. Right-click with you mouse on the Device and **add it to your Project**
5. **Select the device** in the **Project Explorer**
6. Go to the **Inspector and set the values** for each button



Your Stream Deck device and all settings are stored into your VERTEX project file. When you load the Project, just connect the Stream Deck to your PC and go ahead with your work.

## Settings



<b>1</b>	<b>Host System</b>	Defines the host System into your Project <b>where your Stream Deck physically is connected to.</b> Only change when working into a project with Session Member Systems.
<b>2</b>	<b>Label</b>	Set a <b>Label</b> that is <b>displayed as text</b> for your Stream Deck button.
<b>3</b>	<b>Script</b>	Enter <b>Script Commands that should be executed when the Stream Deck Button is pressed</b> Focus with your Mouse and use Shortcut "CTRL + Space"to show a list of all Commands
<b>4</b>	<b>Color</b>	Sets a <b>background color</b> for the Stream Deck <b>button.</b>
<b>5</b>	<b>Icon</b>	<b>Choose an icon for your button. VERTEX comes up with predefined Icons for the most common use cases</b>
<b>6</b>	<b>Vertex Content</b>	Show a <b>thumbnail from a content asset of your project as background of the button.</b> Drag an video or image content with your mouse from Project Explorer to this property field Set Icon to "None". Right-click and open context menu to clear background image
<b>7</b>	<b>Font Settings</b>	<b>Font settings for the button Label</b>



#### **Button Count**

The Stream Deck buttons are counted in rows, starting from top left

## Further Settings

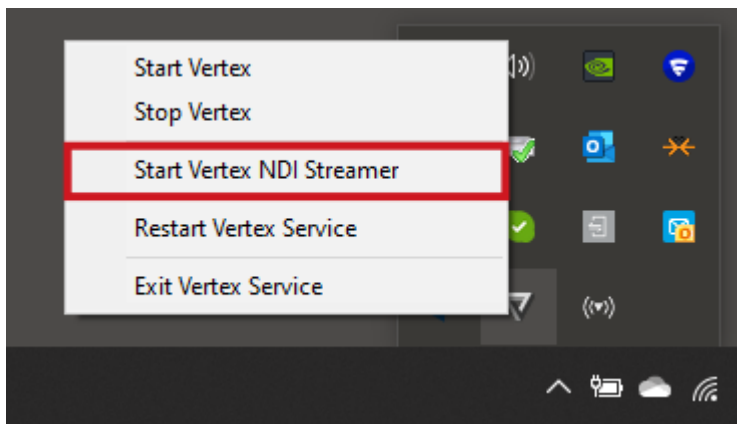
- Select the "Settings"-Tab of the Inspector to set the global brightness of the Stream Deck or to check if the Stream Deck is connected to your PC.

### 5.3.13.5 Vertex NDI Streamer

- VERTEX NDI Streamer is a **multi source NDI toolkit**
- VERTEX NDI Streamer is **capable to stream up to 2 desktop regions and 1 live input** as a NDI Stream
- The NDI Streamer **is free** and fully **works without a valid VERTEX license**
- **In addition, all settings and streams can be controlled remotely out of VERTEX (with a license)**

## Start NDI Streamer

- The VERTEX NDI Streamer comes up in a package with the VERTEX installer
- Just download and install an up-to-date Version of VERTEX
- Right-click with your Mouse on the VERTEX Tray Icon on the Windows taskbar
- Start Vertex NDI Streamer

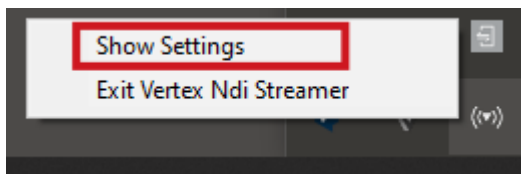


- When the NDI Streamer is started, a new Icon appears into the windows tray bar



## Set up NDI Streams

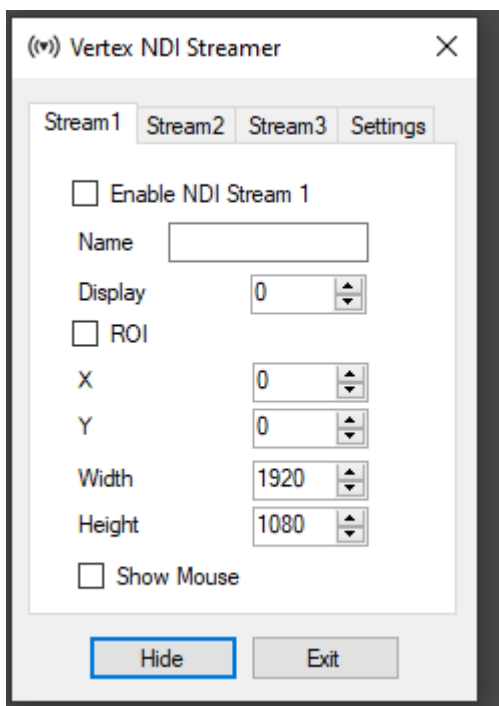
- Right-Click on the VERTEX NDI Streamer tray icon in the windows taskbar
- Select "Show Settings"



- Now you can enable up to 3 NDI Streams from your PC

### Stream 1 + 2

enable you to set up two NDI streams that capture a display (or only a region) of your PC.



#### **Name**

Name of the stream that is shown for all NDI applications

**Display**

Select the display number of your PC that should be sent as NDI stream.

**ROI**

Region of Interest. Default: disabled

Enable to capture only a region of the selected display as NDI stream

Enter pixel coordinates (starting at top left) and width/height to define the ROI

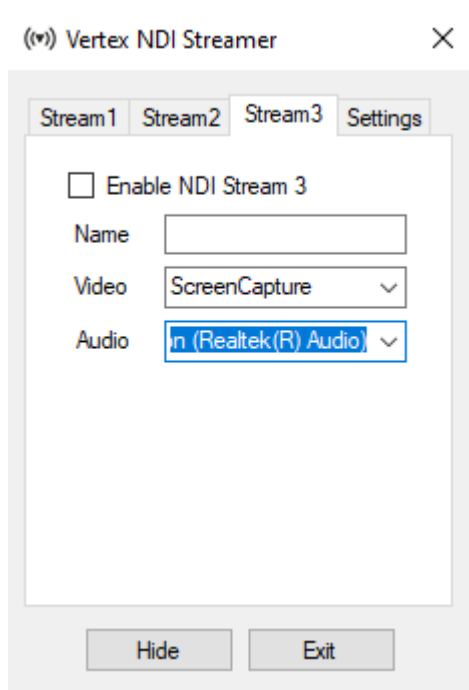
**Show Mouse**

Show mouse pointer in NDI stream

## Stream 3

Stream 3 differs from the first ones. Here, you are able to send a video source (e.g. a webcam) and audio source (like a microphone)

from your Windows PC as NDI stream.

**Name**

Name of the stream that is shown for all NDI applications

**Video**

Select from a list of video sources.

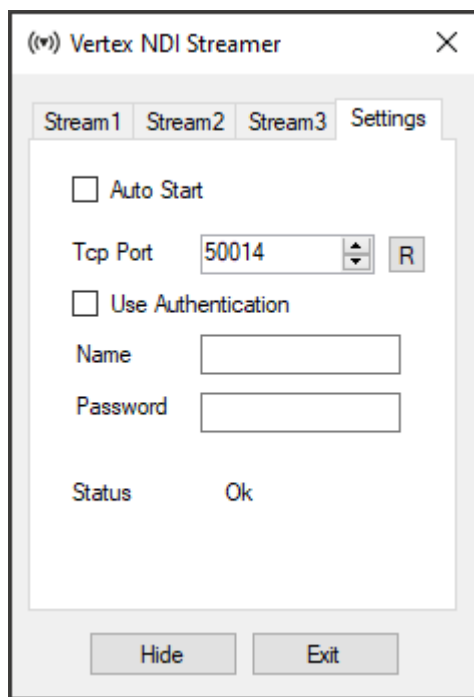
Depending on your PC hardware and configuration the list of available sources may vary



**Audio**

Select from a list of audio sources.

Depending on your PC hardware and configuration the list of available sources may vary

**Settings****Auto Start**

Enable to add the NDI Streamer to Microsoft Windows autostart

**TCP Port**

Change port number for remote controlling the NDI streamer over TCP.

Use the R-Button to restart the TCP Servers after a port change.

**R Button**

**Reset and Restart the TCP Server for all network adapters after you have changed the TCP Port**

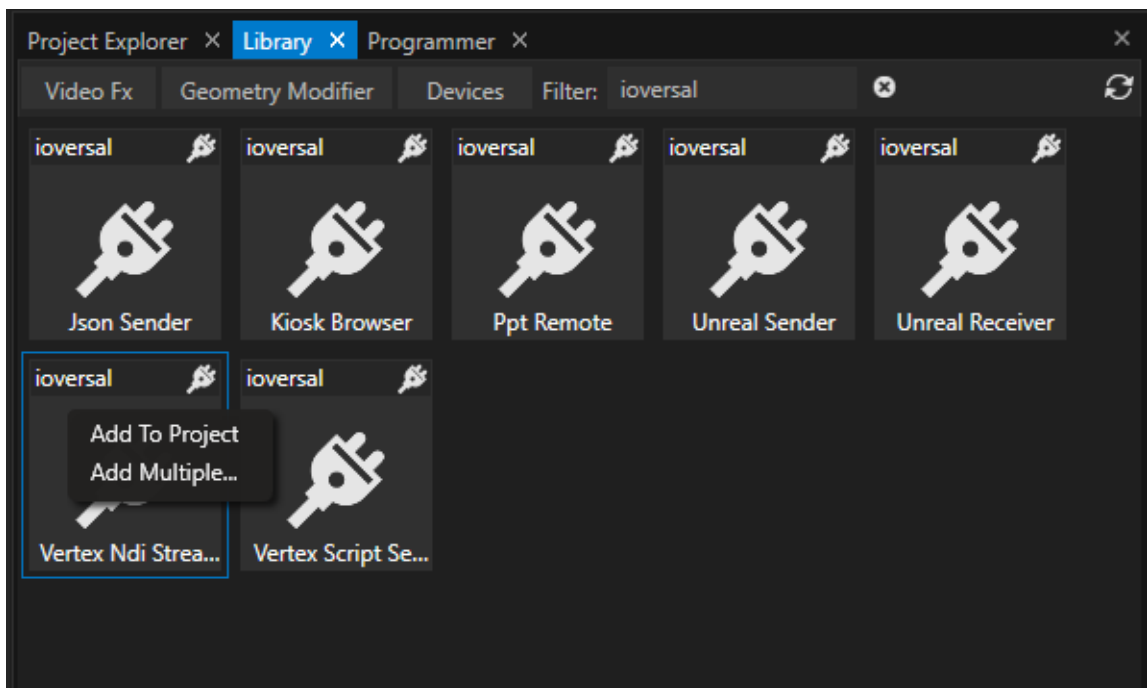
**Use Authentication**

Set User Name and Password for TCP Authentication

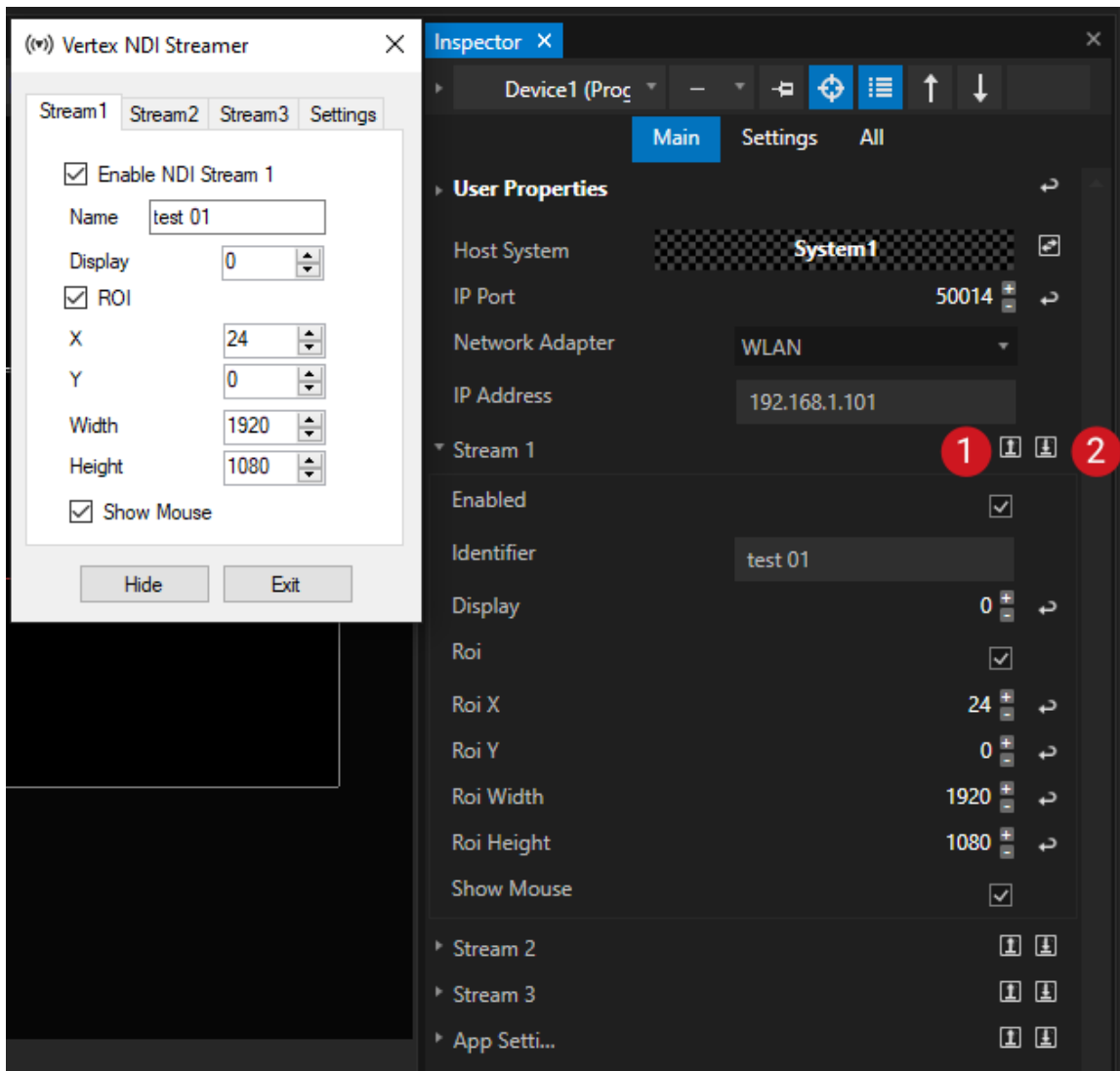
## Remote Control out of VERTEX

- You can **remote control all settings of the VERTEX NDI Streamer** on a PC in your network **out of VERTEX**
- Just add a **"Vertex NDI Streamer" Device** from the [Library](#) into your current project
- Once configured, you have **access to all settings of the NDI Streamer application** and are able to automate your NDI workflow

### Setup



- Open the [Library](#)
- Select **Devices** and filter for **"ioversal"**
- Right-click to the device **"Vertex NDI Streamer"**
- **"Add to project"**
- Select the device in the [Project Explorer](#) and set the initial properties in the [Inspector](#)
- Once the connection is set up, you are able to upload and download all application and stream settings.



Comparison: All settings from the NDI Streamer application (on the left) you will also find into the Inspector when selecting the corresponding Device "Vertex NDI Streamer" (on the right)

### IP Port

Enter here the same IP Port as set in the NDI Streamer application

### Network Adapter

Choose a network adapter from your local VERTEX instance

### IP Address

Enter IP address of the PC on that the Vertex NDI Streamer application is running and should be remote controlled

### Upload Settings 1

Upload all current settings from VERTEX to the NDI Streamer application

### Download Settings 2

Download all current settings from NDI Streamer application to Inspector

## Scripting

As with every Device in VERTEX, you can control an NDI Streamer Device with a [Script](#).  
For the following examples the NDI-Streamer Device was added as "Device1" to a project

### Enable and disable Stream 1 on the remote PC

```
Device1.Settings.Stream1.Enabled.Value = true  
Device1.Settings.Stream1.Enabled.Value = false
```

### Show Mouse Cursor in NDI Stream 1

```
Device1.Settings.Stream1.ShowMouse.Value = true
```

## 5.4 Content Types

- The VERTEX encoding engine is **based on FFmpeg** - the most common and popular **video/audio codecs and file formats** are supported
- VERTEX offers you advanced options to **optimize Image Sequence Playback**
- Furthermore there are **many other content types** like HTML, Live Inputs or Generative Patterns
- There are **various options** how to import content including the **import of whole directories**

### Import Content

- There are various ways for [content import](#) in VERTEX. [Choose the way](#) that fits best to your workflow and your project.
- You can also [import whole folder structures](#) directly into VERTEX. And for the import of [Image Sequences](#) there are additional settings.
- All options are summarized in the separate chapter [Import Content](#). There you will also learn some things about [advanced settings for content import](#).
- [Watchfolders](#) are a tool to manage and automate content import

### Types of Content

VERTEX supports a bunch of different content types. From videos, audio and images up to generative content like test pattern or even live inputs, if available on your hardware.

To get more information go ahead with the detailed-chapters below:

#### Basic Content

- [Video](#)
- [Images](#)
- [Audio](#)
- [3D Objects](#)

#### Special or advanced content types

- [Image Sequence](#)

- [Live Input](#) like Cameras, Input Cards, NDI
- [Procedural Content](#) like Testpatterns, Solid Colors or Gradients
- [HTML-Content](#)
- [Text Content](#)

## File formats from 3rd- Party applications

- [Powerpoint Files](#)
- [Adobe Photoshop-Files](#)
- [Notch Blocks](#)

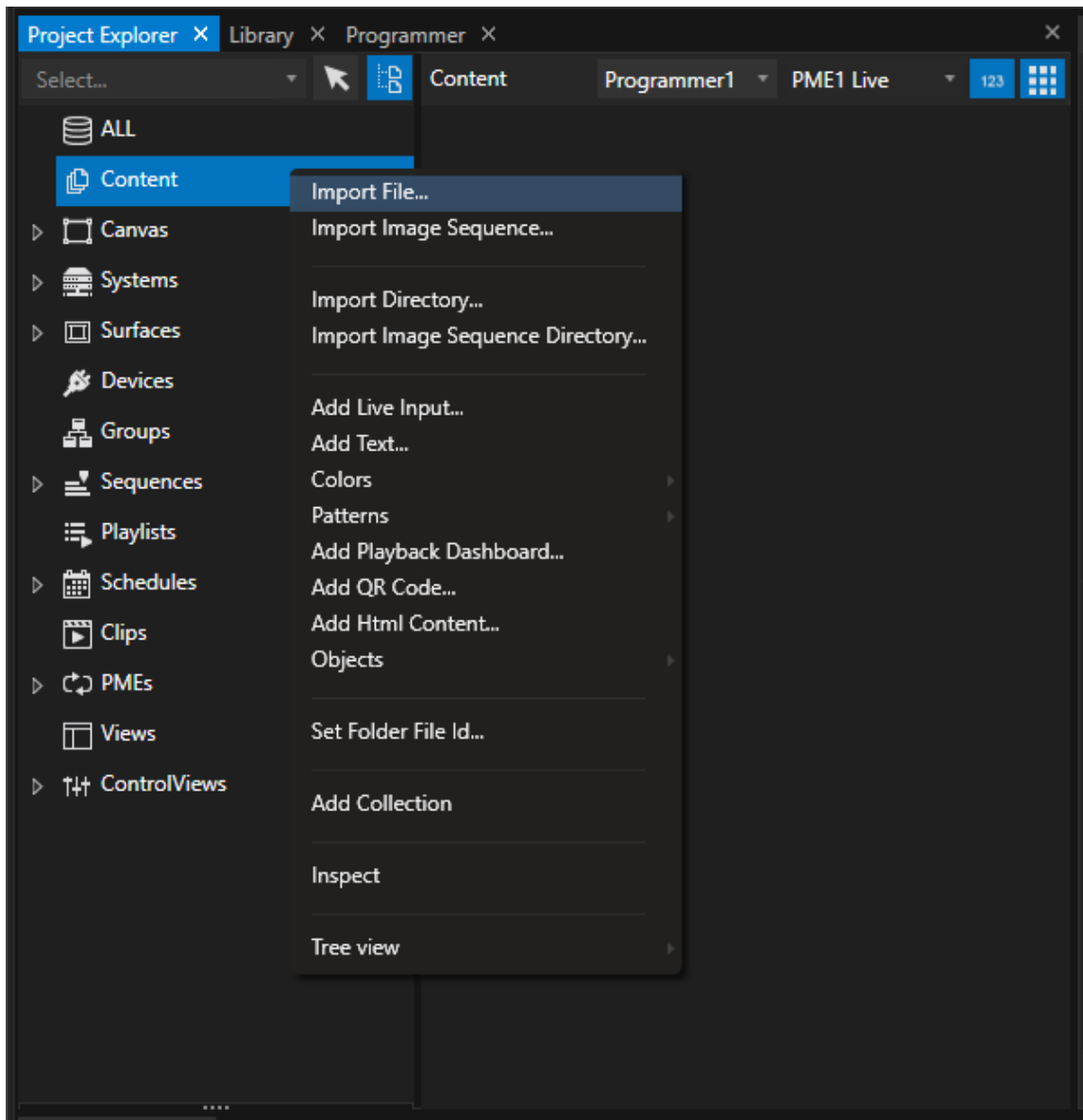
### 5.4.1 Import Content

- There are **different options to import content** into VERTEX: using a shortcut, with the help of menus or just with drag and drop
- Import **one single file, multiple files or a whole directory** including all files and folder structures
- There is an additional option to **import image sequences** or a directory of image sequences

## Various ways to import

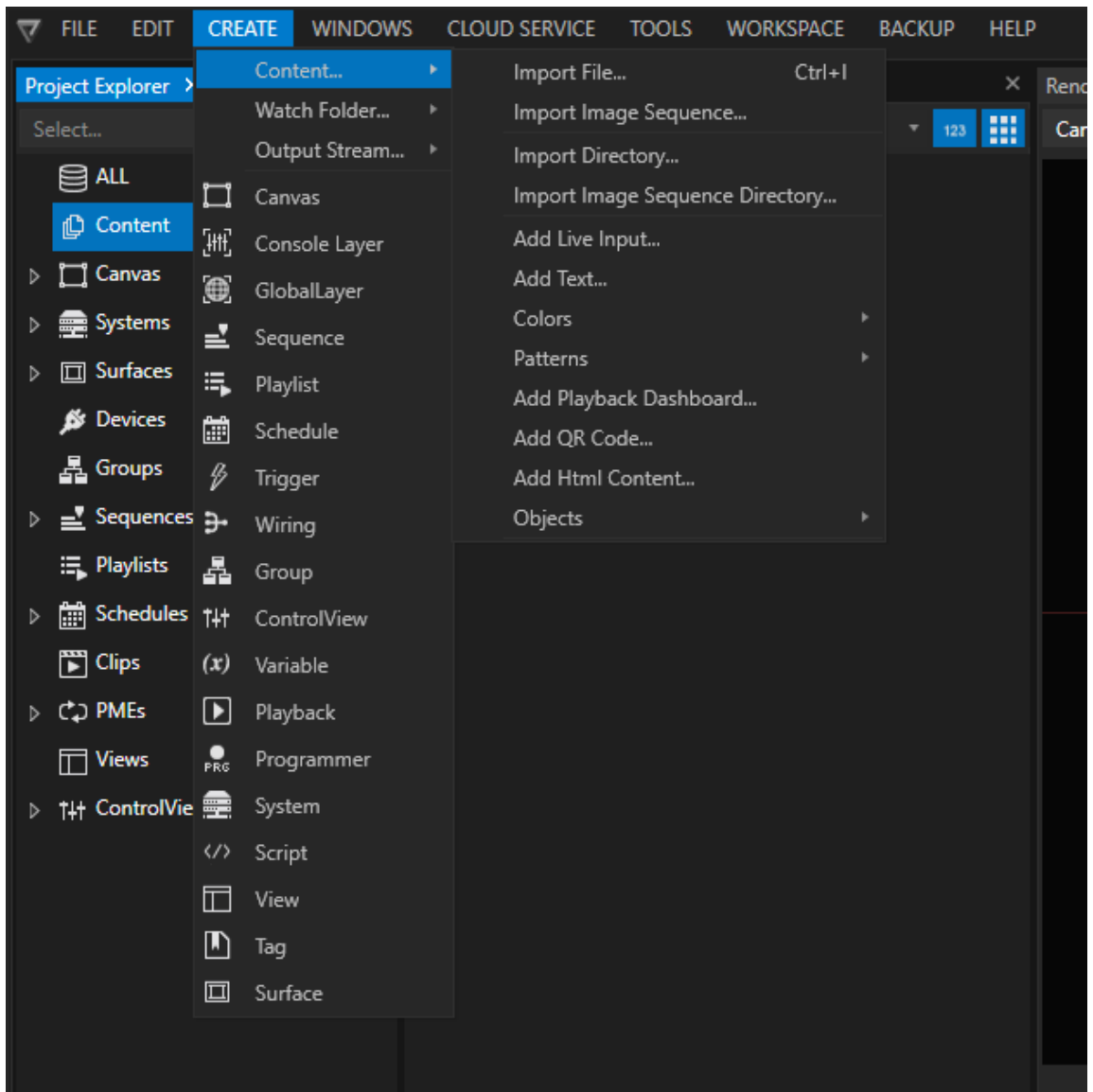
### 1. Context Menu in Project Explorer

Right-click with your mouse on the "Content" section in the Project Explorer  
Choose your [import option](#) there.



## 2. Main Menu: "Create"

Go to the Main menu on top, fold out the "Create" Tab and select your import option there



### 3. Shortcut CTRL+I

Focus the Project Explorer and press the keys CTRL+I

The Windows Explorer opens. Choose your files there and confirm

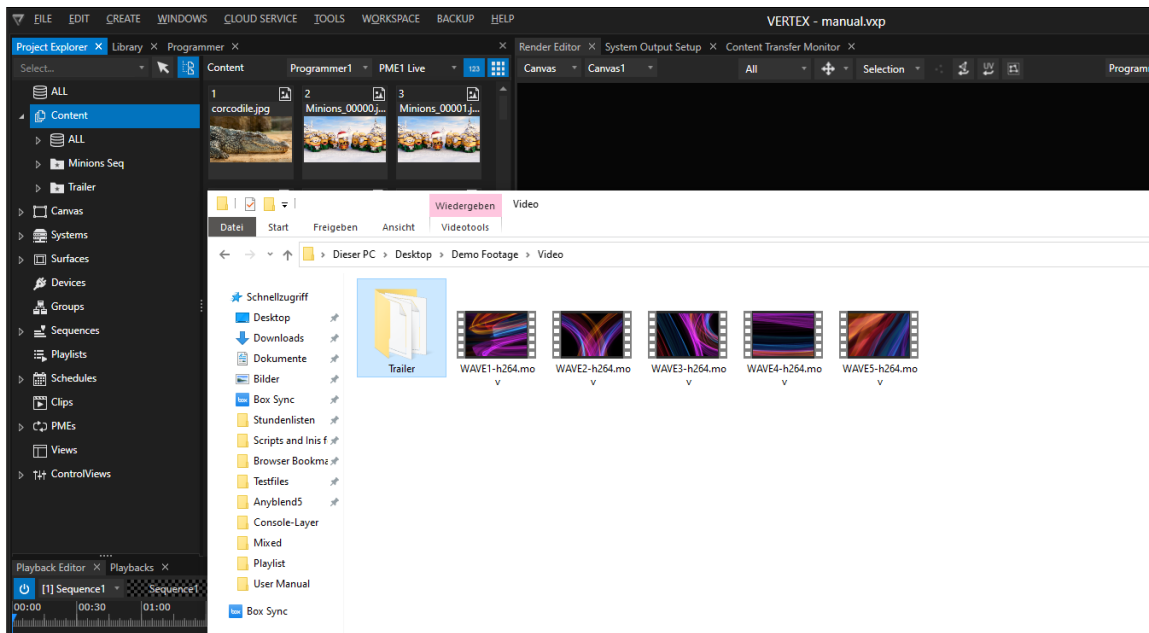
### 4. Drag and drop from Windows Explorer to Project Explorer

VERTEX allows you to add content simply via drag and drop from Windows Explorer.

Select your file into the windows file explorer and drag it to the Project Explorer or directly to your sequence

This also works for simple folders without subfolder.





## Directory or File(s)

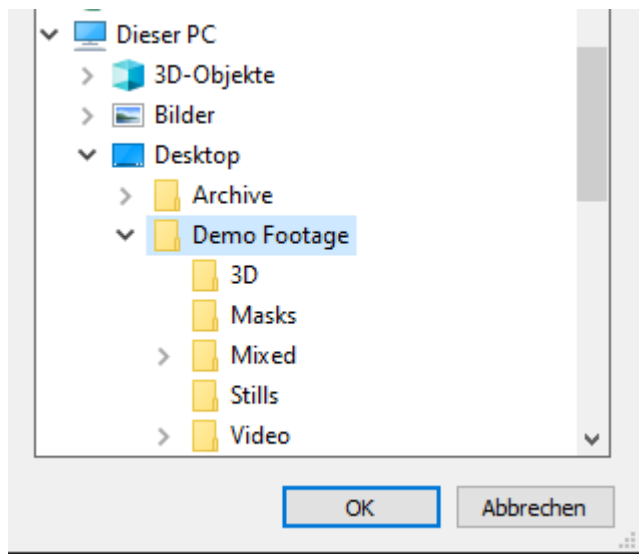
*You can choose whether you want to import a single file, several files or a folder structure.*

### Import one file or several files

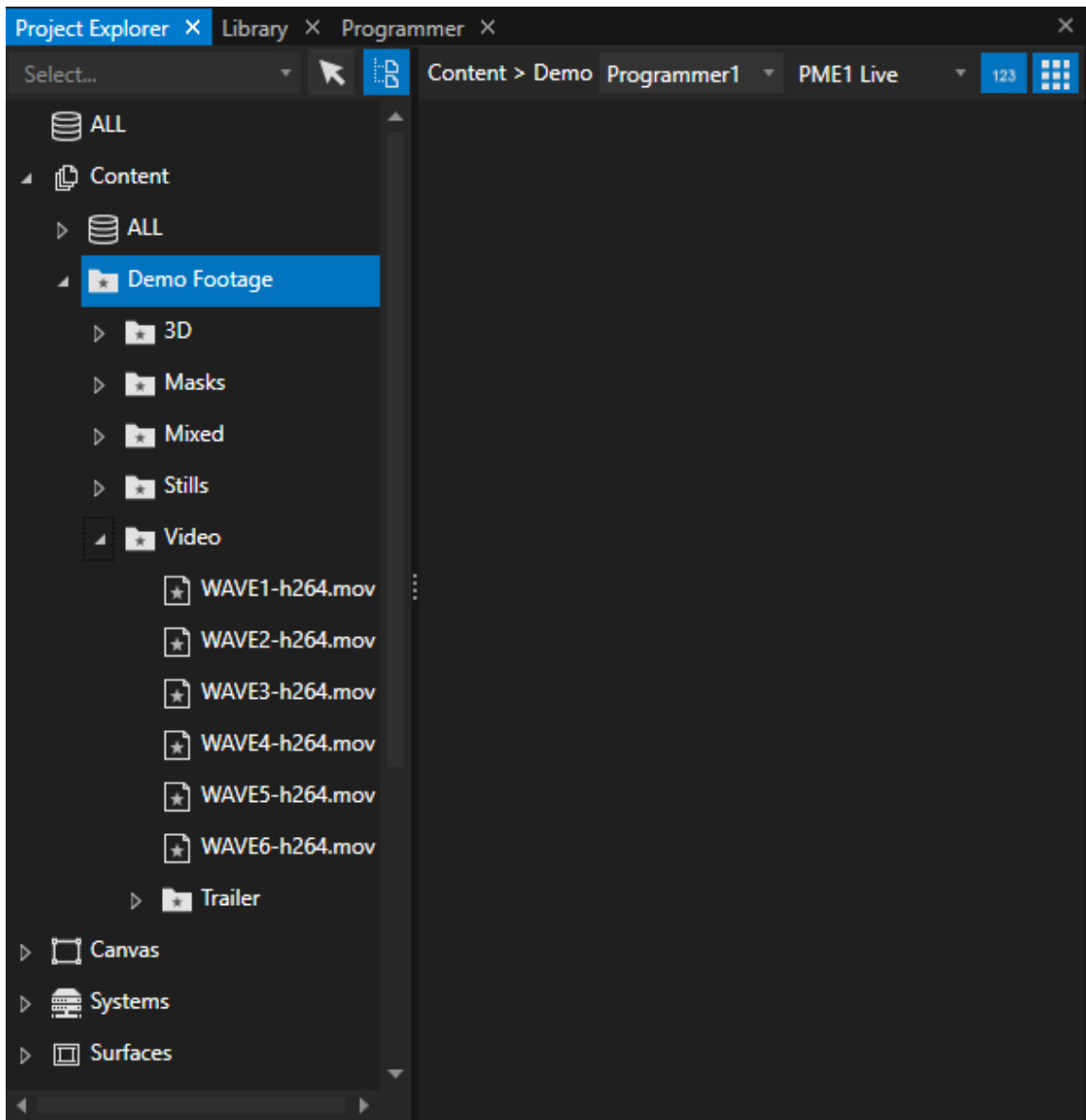
- Select "Import File"
- The Windows Explorer opens
- Select one or multiple files there into the same directory

### Import Directory

- Select "Import Directory"
- The Windows Explorer with a Directory tree opens



- *Choose a directory*
- *The directory, all subfolders and all content is imported*
- *Subfolders are taken over as collections using the same structure*



## Import Image Sequence

*Because of some settings there is an extra option to import Image Sequences.*

*The Image sequence import is available as menu entry in the Project Explorers context menu or the Main Menu.*

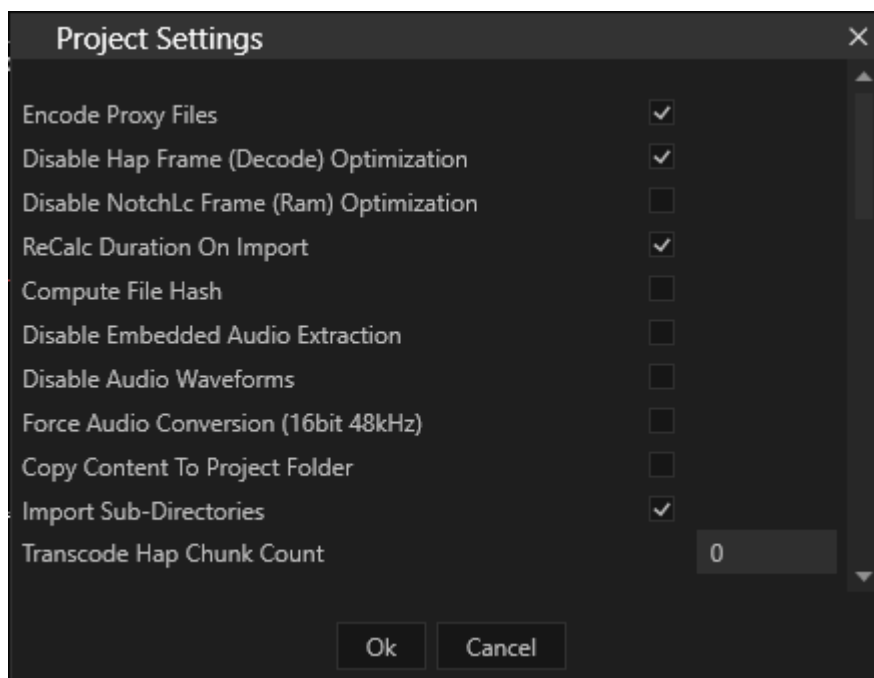
*Learn more about import and options for Image sequences here: [Image Sequence](#)*

## Settings

Initially, there are some basic import options set in VERTEX that should work for the most projects. Beyond that, you can customize the behaviour during import.

### Project Settings

- globally define details for the file import.
- switch the [Proxy file encoding](#) on/off
- set special options for codecs (NotchLC and HAP)
- select what should happens with Audio (Disable embedded Audio extraction, disable audio waveforms in content tiles, force an audio conversion to 16Bit/ 48kHz)
- define the data handling (should the original content be copied to the VERTEX project folder? Should Sub-Directories imported...?)
- define the [Content Sync](#) behaviour for a session with [multiple systems](#)

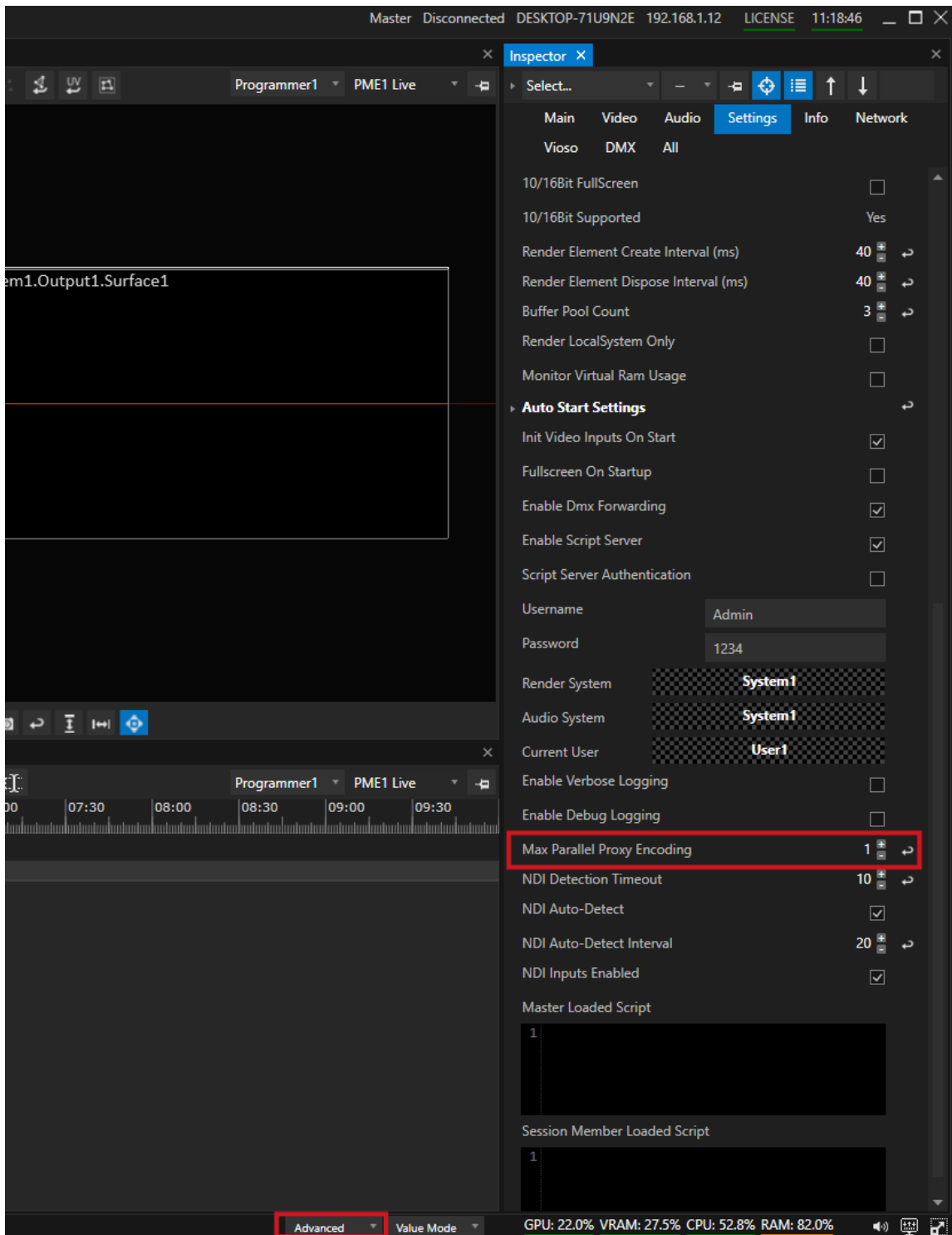


Go to the Main Menu on top, open "Edit" tab there and select "Project Settings"

## System Settings

For every System into your project you are able to define how many [proxy files](#) should be calculated in parallel (Max parallel Proxy encoding)

Depending on your hardware, you are able to increase the number to speed up the encoding and the import. As default, the value is set to 1. One file after another will be encoded to limit the increased hardware load.



## Advanced

There are some advanced options to handle or control content

### WatchFolder

Define and observe Windows folders. A content import is automatically triggered if content was changed or added into this WatchFolder.

As Option, you are able to let VERTEX automatically create Playlists or Sequences based on the WatchFolder content

[Learn more about WatchFolders](#)

### Folder and File ID

Folder and File IDs allows you to simply access content and folders by an value based ID.

This is necessary when e.g. working with DMX and Console Layers

Learn more about a [Console Layer](#) or the [Folder and File IDs](#)

## 5.4.2 Video

- The Encoding Engine in VERTEX is based on FFmpeg. Basically, VERTEX is able to import **every codec and file container that is supported by FFmpeg**.
- Depending on your hardware setup, **not every codec is recommended for every use case**
- **VERTEX supports high quality codecs like Apple ProRes, HAP/HAP-Q and Notch LC**
- VERTEX automatically extracts and resamples embedded Audio

## Supported Formats and Codecs

- VERTEX is **based on ffmpeg** to read video data - basically **all video formats from ffmpeg are supported** (e.g. MPEG4/H264, MPEG 2...)
- Playback of high quality codecs like **Apple ProRes, HAP/Hap-Q or Notch LC** is supported
-



### Choosing the right codec

The choice of a codec depends on your projects conditions, the resolution and your quality requirements. A general statement is difficult and also does not represent the full range of projects. The **choice of a codec** also depends on your hardware configuration. Some codecs require more GPU power, others require more CPU power or a fast hard disk. For **high quality playback** we recommend to use Apple ProRes, HAP, HAP-Q or Notch LC. VERTEX comes with a **native integration of HAP and Notch LC**. Most parts of the encoding load is shifted to the GPU. The playback performance for especially high resolution files is great.

If there are any **questions** about codecs, please write us an email to [support@ioversal.com](mailto:support@ioversal.com)



### 10 Bit Playback with Notch LC

VERTEX supports 10 Bit Playback with the NOTCH LC codec. This codec is free. "It brings the equivalent of 10bit accuracy in a scrubbable codec that is extremely fast to encode and decode, with a compression ratios of around 5:1" (Source: [Notch LC Website](#))

## Related Links



### Notch LC

[Official Website of the free Notch LC Codec](#)

### HAP

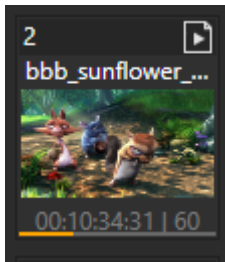
[Official Website of the free HAP Codec](#)

## Proxy Files

- For each video content that you add to your project, VERTEX **automatically calculates proxy files** in a **small** and a **middle quality** and resolution.
- The **progress of this encoding** is displayed **with an orange progress bar** into your content tile
- Into the [Render Editor](#) you are able to **switch between original file and proxy file**.

## Encoding and Status





- The **orange progress bar** into the content item in Project Explorer shows you the **status of the encoding process**
- By default, the **amount of parallel encoding jobs** into a VERTEX project is **set to 1**. When import multiple files, the encoding will be done one by one.
- **All Proxy Files are saved into your VERTEX Project Folder**. When the Encoding Process is finished, they **are distributed in background to all other VERTEX Systems** into a Project.



When VERTEX is creating the proxy file of your video content, the CPU usage is higher as usual.

When the process is finished you should observe a decrease of your PC's CPU usage.

## Settings

- For **each VERTEX System** you can set the **number of parallel encoding jobs** in the Inspector.  
Select your System into Project Explorer and navigate to Inspector (All-Settings Tab)
- For each **VERTEX Project** you are able to **globally disable the encoding of proxy files**  
Go to Main Menu -> Edit -> Project Settings
- For **each VERTEX System** into your current VERTEX project, you can **disable the Content Sync for Proxy Files**.  
Select your System into Project Explorer and navigate to Inspector (All-Settings Tab)



*Proxy Files are saved into your project folder.*

*For each Content a Folder with an internal UUID is created into the project files subfolder Data\Content*

## Embedded Audio

- **Embedded Audio of a Video file is extracted and resampled into a separated .wav file during import**
- **The audio file is stored into the VERTEX project folder.**
- For you as user, there is nothing special to keep in mind - When working with this video file, VERTEX always will automatically assign the resampled Audio track to your video. But if needed, you are able to change to audio track of your clip container.

## Settings

- Audio Extraction can be **globally disabled for a VERTEX Project** into **Project Settings**.  
Go to Main Menu -> Edit -> Project Settings

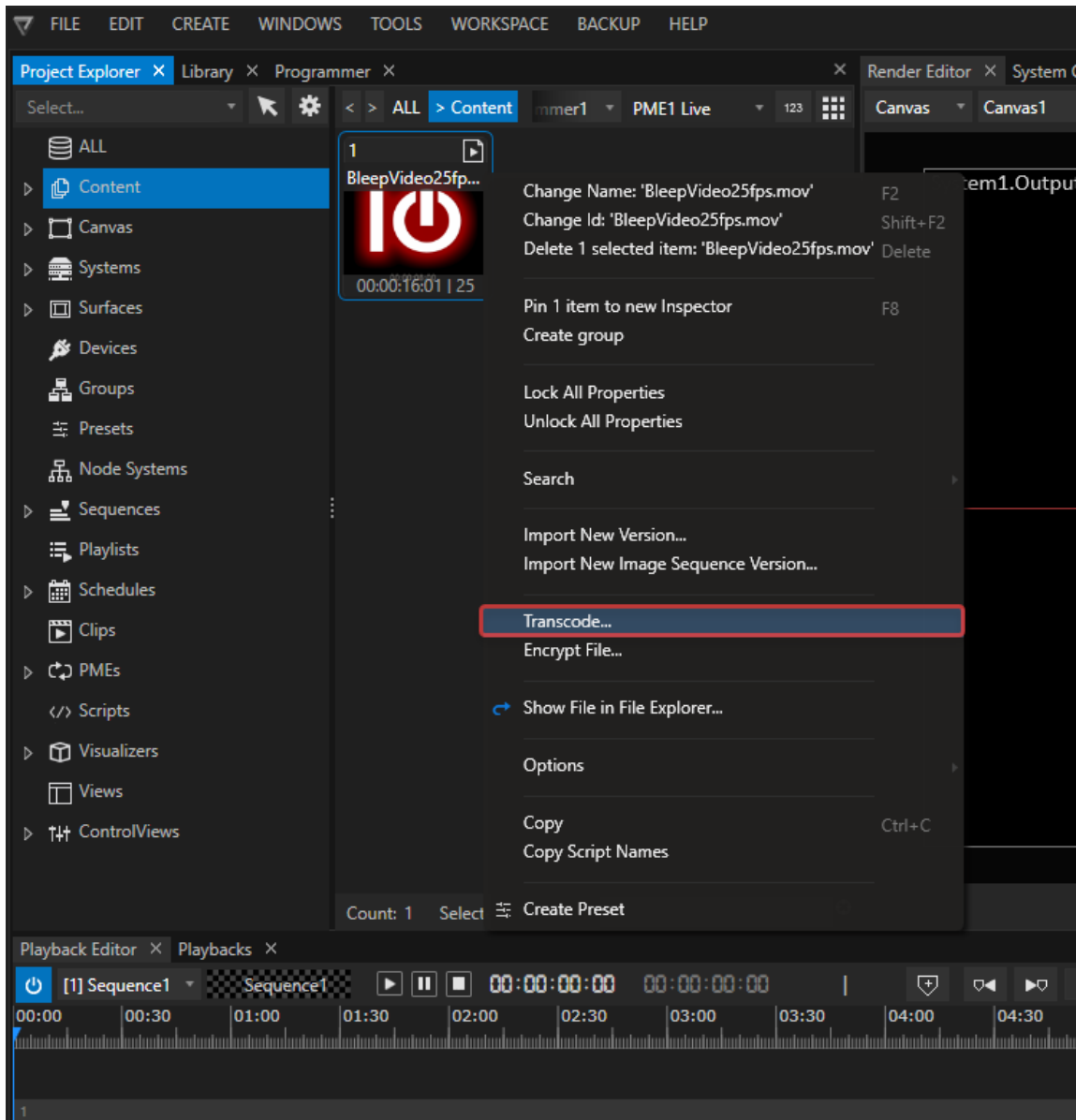
## Transcoding Presets

- Transcoding for the **most common codec types** is available with a **simple right-click in the Project Explorer**
- No complicated settings are needed: VERTEX **offers you Presets with the ideal configuration** for best performance in VERTEX
- Choose weather the new file should be imported as **new content** or as **new version**

## How to

- select the already imported **Content** into the **Project Explorer**
- **Right-Click** on this Content and open the **context menu**
- select **Transcode..**
- The **Transcode Settings** dialog will open (c.f. below)

- Do your **settings** there and **confirm**
- Transcoding will start - including proxy files. The status is shown with a yellow progress bar in the content items tile

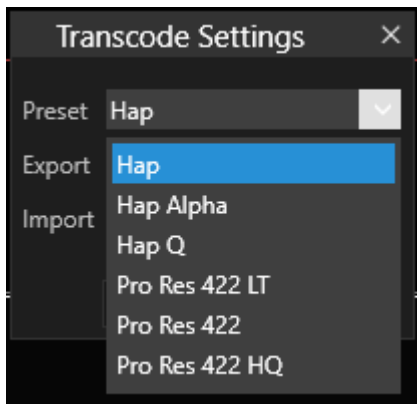


## Transcode Settings

### Presets

Choose from a list of specific codecs.

The presets for the specified format already have the optimal settings for a playback in VERTEX



#### **File Sizes**

Depending on the source file, its resolution and length, new transcoded Hap files (especially hap q ) and ProRes files could have bigger file sizes! Ensure that there is enough space on your project data drive.

#### **Export**

Select between the options "**Audio&Video**" (default setting) and "**only Video**"

#### **Import**

There are two different options how to import the new and transcoded video into your current VERTEX project:

1. As **new content** item

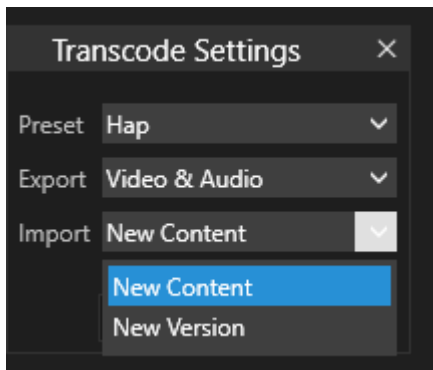
The transcoded video will be added as a new content item into [Project Explorer](#)

The new transcoded file and its proxy files are stored into the VERTEX project folder.

2. As a **new version** of the **current content item**

The file will be transcoded into the chosen preset and is accessible [as new version of the current content item](#)

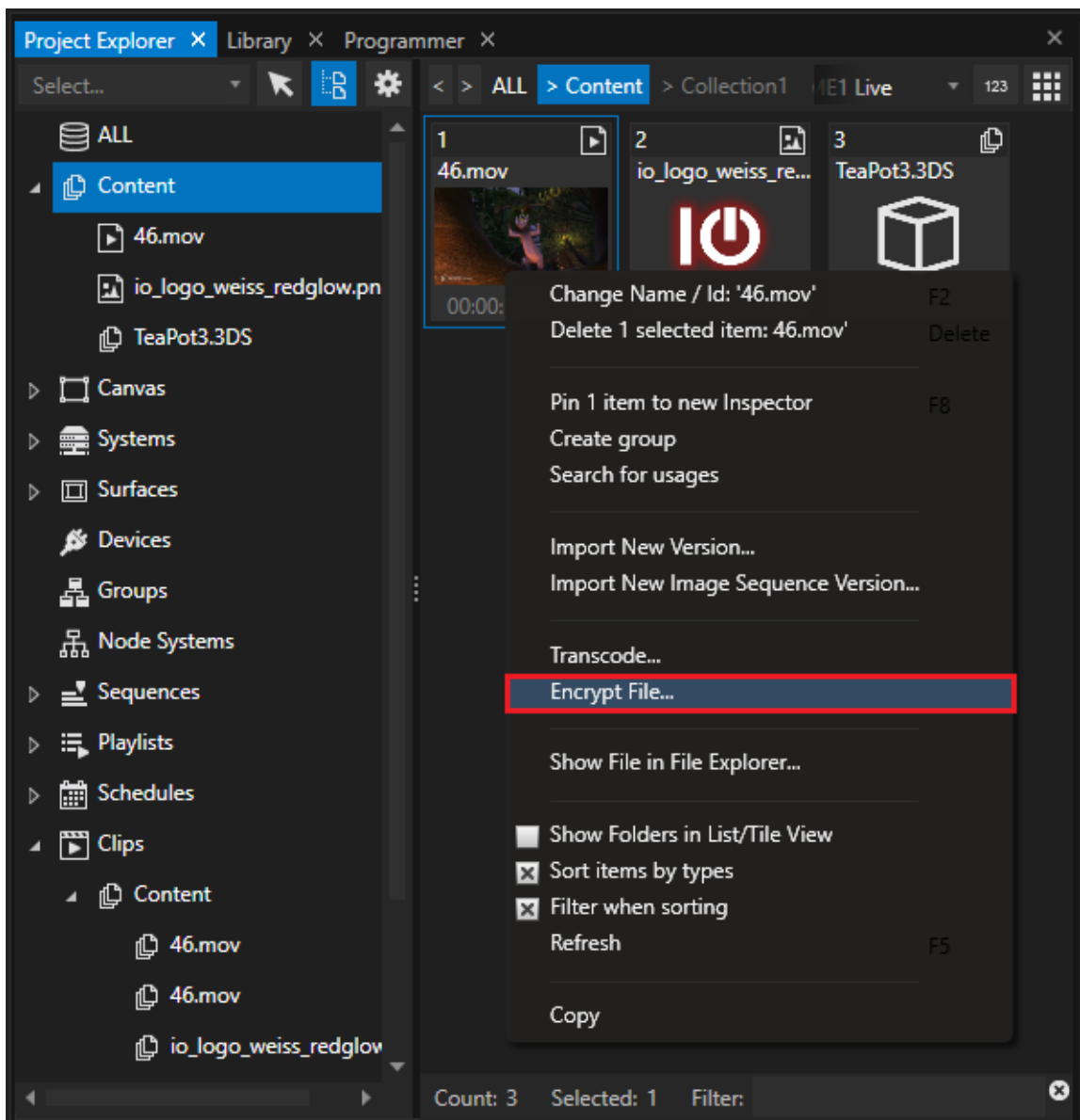
The new transcoded file and its proxy files are stored into the VERTEX project folder.



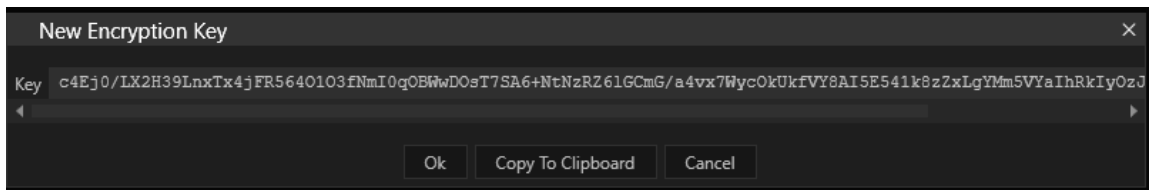
## Content Encryption

- Protect your intellectual property with an **integrated Digital Rights Management (DRM) tool**.
- Video files (.mov - Quicktime container) can be **encrypted directly inside of VERTEX**
- Select between **different encryption modes**: a key, lock it to a license or dongle and/or fit it with an expiry date.

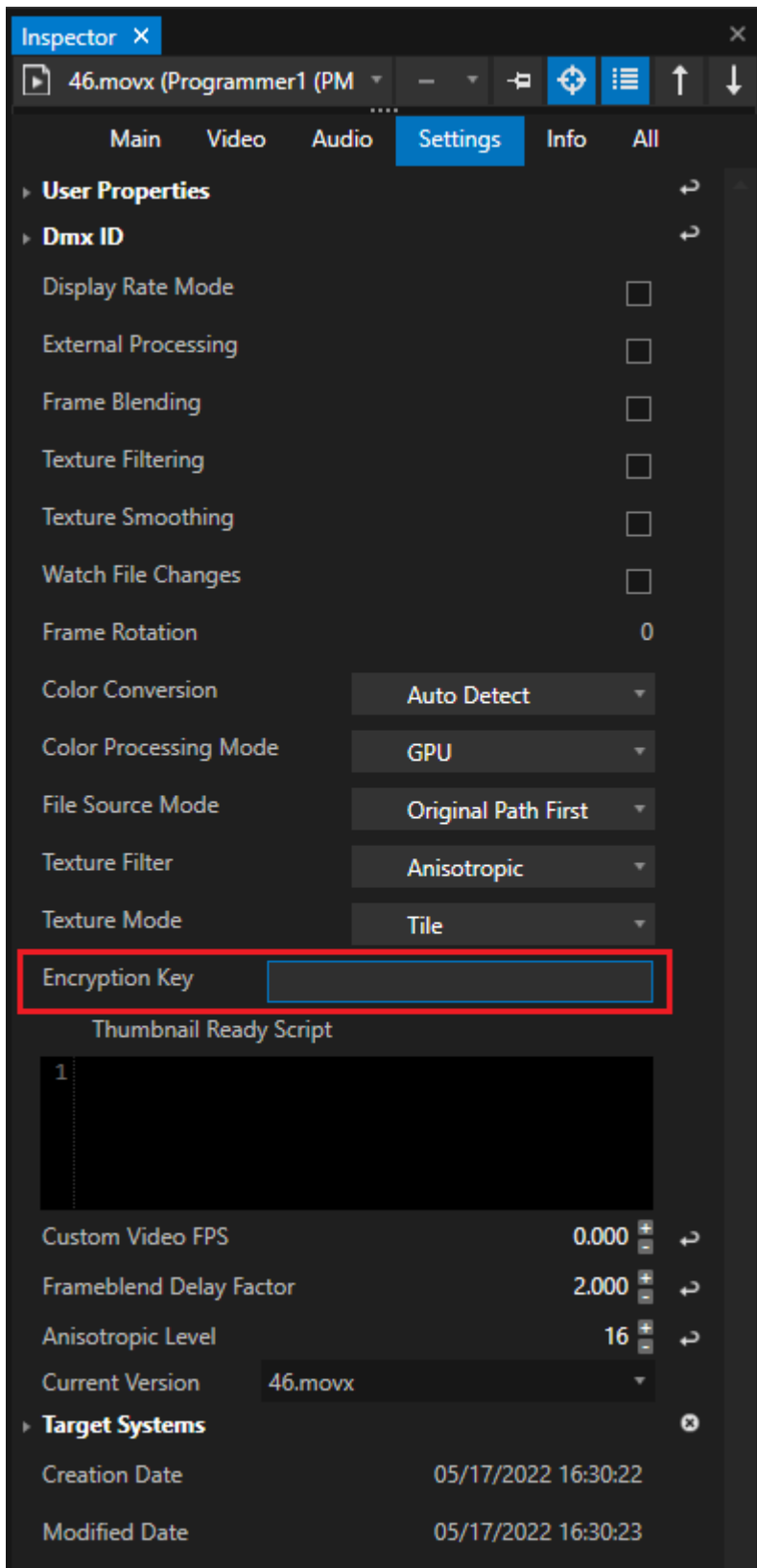
### How to



- Rightclick on a Video file into project Explorer
- Choose "Encrypt File" from Context Menu
- The "Encryption Settings" Dialog will open
- Do your settings and choose mode (see below)
- Copy the encryption key, confirm



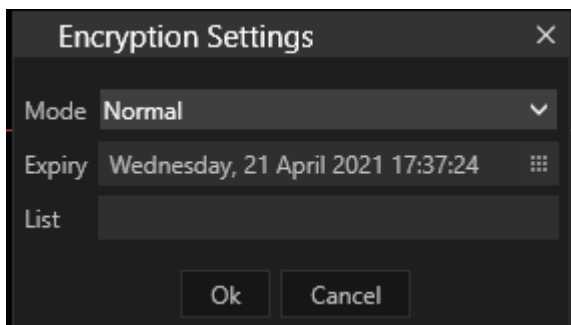
- VERTEX will transcode your video into a new .movx file
- If not automatically imported into project explorer open context menu on the original file and choose "Show file in File Explorer"
- Import the .movx file again
- select the file into Project Explorer
- change to Advanced Inspector Mode
- Search for Encryption Key in Inspector and enter the key





- Refresh Proxy and Cache files: Right-Click - Context Menu on the encrypted content item in Project Explorer

## Settings and Modes



### **Normal:**

File is encrypted with a passkey. The passkey is generated after clicking OK

### **Expiry Date:**

The File receives an expiry date, for how long the file can be opened

### **License list:**

The file will only run on systems with a certain license key. The license key can be entered into the list field

### **Dongle:**

The file can only be opened if the dongle plugged in during encryption is plugged in during playback

## 5.4.3 Image Sequence

- VERTEX converts image sequences optionally in a **proprietary ioversal file format to ensure performance stability during playback.**
- Of course you are also able to use the original files as source format.
- The **playback performance** depends very much **on the chosen file format** and **especially your hardware.**
- There is an option to import image sequences as **whole directories with subfolders.**
- **TIFF is natively supported in RGB 8/16 bit RGBA 8/16 bit.**

## Supported Formats

*The following formats work with VERTEX.*

*Their playback performance varies, especially when working with compressed formats.*

*BMP, TIFF, TGA, JPEG, PNG, DPX, GIF*

*We strictly recommend to test format and playback performance with the [VERTEX trial version](#) beforehand.*

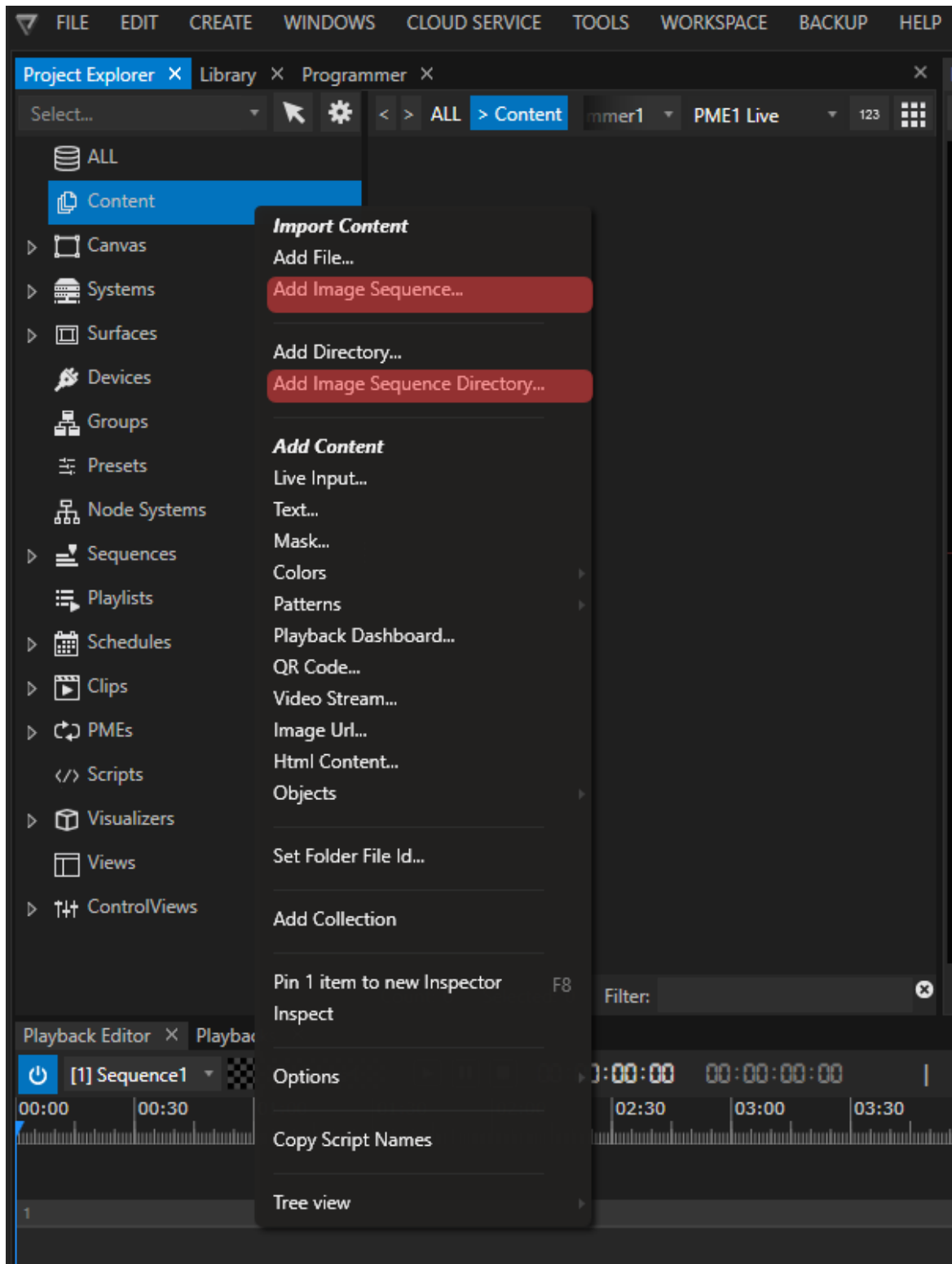
*16 Bit Playback works with TIFF images.*

*Also DPX supports 16 Bit, but is not yet tested in all cases.*

## Import and Playback

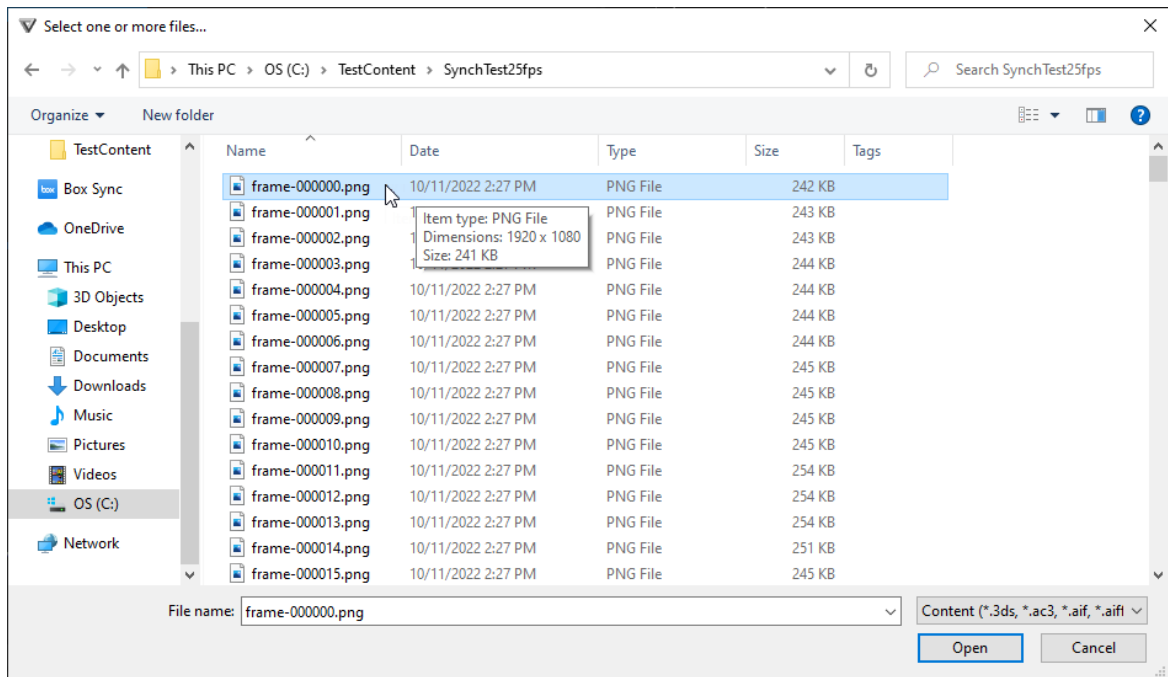
### Import

*You can import either an image sequence or a whole directory structure of image sequences into VERTEX:*

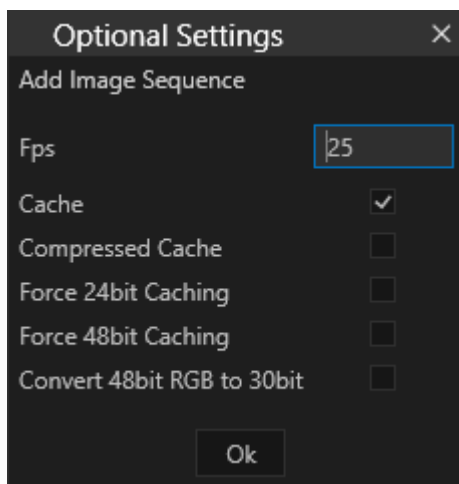


Go to Project Explorer > Content > right-click context menu or Main Menu > File > Add Image Sequence (or Image Sequence Directory).

After giving the import command, select the first file of your image sequence:



Because importing image sequences is different from any other media type, you will be presented special options after the initial import command:



## Fps:

Sets the frame rate for your image sequence. When proxy file encoding is enabled, this frame rate is used to encode the image sequence to proxy file videos.

The frame rate is used for your content asset in the project explorer. Go to the inspector to change the settings.

## Cache:

**Enable** for VERTEX to convert the imported image sequence into a proprietary .iovi format and store it in your project folder.

**Advantage:** optimized file processing, optimized data handling, optimized algorithm - ensuring the best

playback performance

**Disadvantage:** the conversion of tga, and dpx files costs time and processing power.

**Disable** if you want to **import the original file format.**

Depending on the file format, playback performance can vary.

The system where the import was done uses the original files are for playback.

In a multi-client session the files are being copied by default to all session members' project folders (content sync).



Please keep in mind that your project folder has to be located on a very fast drive with enough disk space.

Double check this also for all other VERTEX systems when working in a multi-client session.

Importing an image sequence directory can fill up your drive quickly with all the data created and copied into your project folder(s).

## Compressed Cache

**Default Setting:** Disabled.

If enabled, Data into ioversal Cache format (.iovi) will be compressed - (only binary data from the whole file - no effects on image quality)

**Advantage:** Each file of the image sequence will be about 20% smaller. The amount of data that has to be read from the hard drive becomes smaller.

However, data compression has to be reverted by the CPU - leading to a higher CPU load during playback.

Recommended when playing big or multiple image sequences on a system with high power CPU.

**Recommendation for the best work routine:** have your content creators export and deliver your content in a lossless PNG file format.

## Force 24 Bit Caching

**Default Setting:** Disabled

Enable for the ioversal cached format - all data is converted to 24 Bit (R G and B @ 8Bit)

## Force 48 Bit Caching

**Default Setting:** Disabled

*Enable for the ioversal cached format - all data is converted to 48 Bit (R G and B @ 16Bit)*

## Convert 48 Bit RGB to 30 Bit

**Default Setting:** Disabled

*Enable for the ioversal cached format - all data is converted to 30 Bit (R G and B @ 10Bit)*

### 5.4.4 3D Objects

*The documentation is work in progress and will be updated step by step*

*Until then: Please drop us an E-Mail with your "How-to-do-this-in-VERTEX" question to [support@ioversal.com](mailto:support@ioversal.com)*

## Supported Formats

\* FBX

\* 3DS

### 5.4.5 Images

- VERTEX **supports all major image file formats like JPEG, GIF, TIFF, PNG and BMPs**
- **Alpha Channels and transparency** are supported
- During import all formats are rasterized ( for e.g SVGs or EPS files)

### 5.4.6 Audio

- VERTEX encoding is **based on FFmpeg** and therefore supports a **great variety of audio codecs**.
- When audio is imported, VERTEX **converts all audio formats to a 48kHz/16Bit WAV file**. The new files are stored in the project folder.
- **Multi-channel audio** is also supported.

*Audio files that come in VERTEX' default format (.wav with 16bit /48kHz) are not converted during import.*

*You can, however, force VERTEX in the [Project Settings](#) to automatically convert every audio file that is imported - even if it has already got VERTEX's native format.*



Although each audio file is converted to an industry standard of 48kHz/16Bit -even mp3 audio -, we recommend to only import uncompressed audio for best results in sonic quality.

## 5.4.7 Powerpoint

- VERTEX is able to import Powerpoint presentations as PPT or PPTX File
- All included Slides are accessible as a PNG File into VERTEX
- Videos or Notes are accessible as child elements of the slide they are placed
- Translations between Slides and Animations are not supported

## Supported Powerpoint Formats

- ppt
- pptx

## 5.4.8 Notch Blocks

- VERTEX supports **native Notch Playback**
- Notch Blocks are handled into VERTEX as any other content element: Import a Notch Block into the Project Explorer, drag it to your timeline and create a Clip Container and set the exposed/auxiliary parameters in the Inspector
- For Notch playback you **require a Notch Playback License/Dongle** and an **installed Codemeter runtime** for license management

## How to prepare VERTEX

- Check out the prerequisites for a notch Playback License on [notch.one](https://notch.one)
- Follow the Notch manual for License handling and install the Codemeter License runtime
- Plug In your Notch Dongle and start VERTEX



### **Notch Playback License**

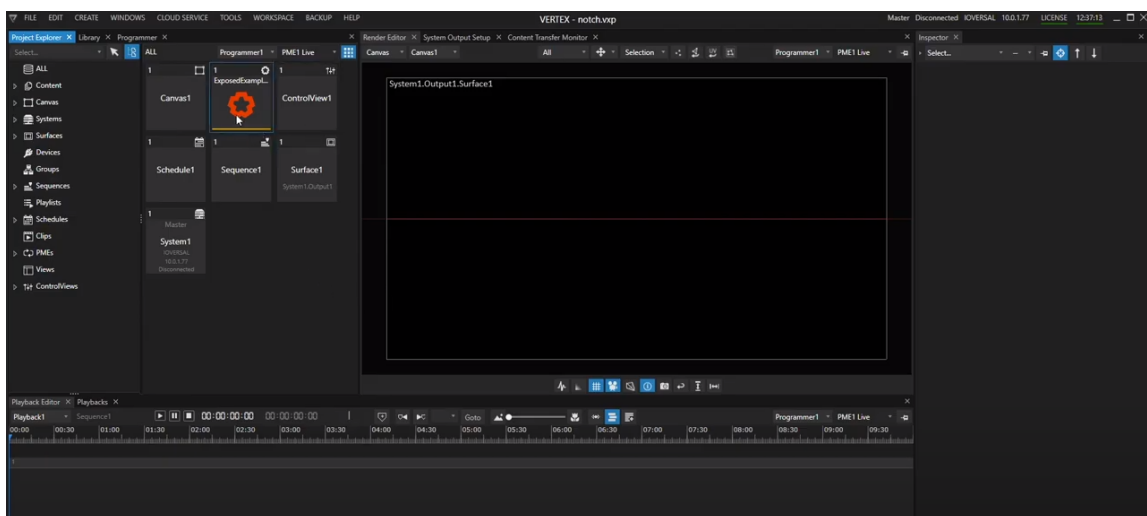
To display Notch content into VERTEX, a Notch Playback license on a specific Notch dongle is needed.

This license has to be purchased at Notch. To run this license, a third party license runtime is necessary.

There is no demo option - if no license is installed or the necessary Codemeter runtime for license handling is missing, an error message will occur. no notch content is displayed

## Working with Notch Content

### Import Notch Content

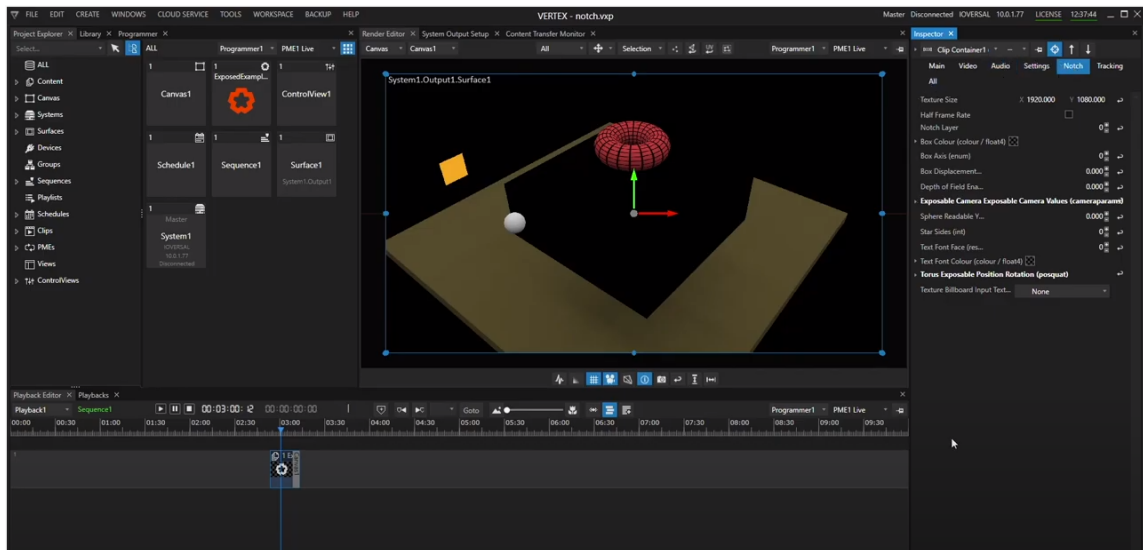


- Notch Blocks are handled as any other content asset into VERTEX.
- after the import there will be a short background processing (orange progress bar on content tile)
- You are able to use Botch Blocks as any other content. settings are made in the Inspector

### Notch tab into Inspector

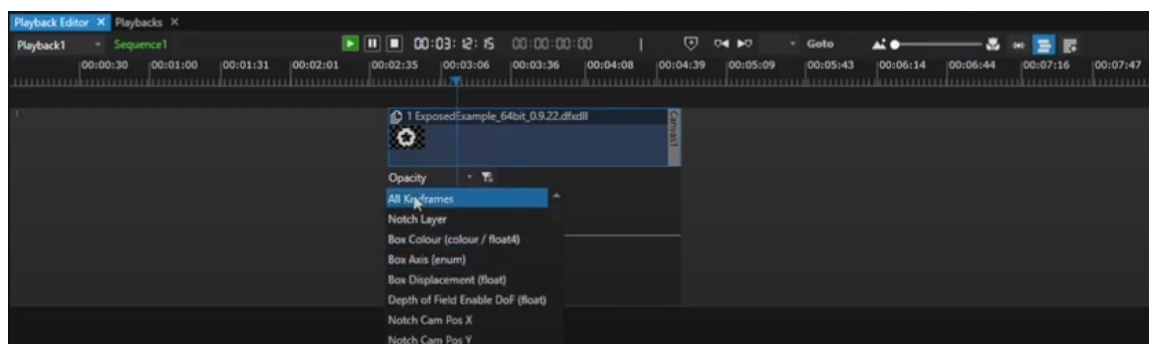
- For Clip Containers with Notch Blocks, there is a tab called "Notch" in the Inspector
- The "Notch" tab shows all possible settings, Notch layers or texture settings that are available in VERTEX for this block.





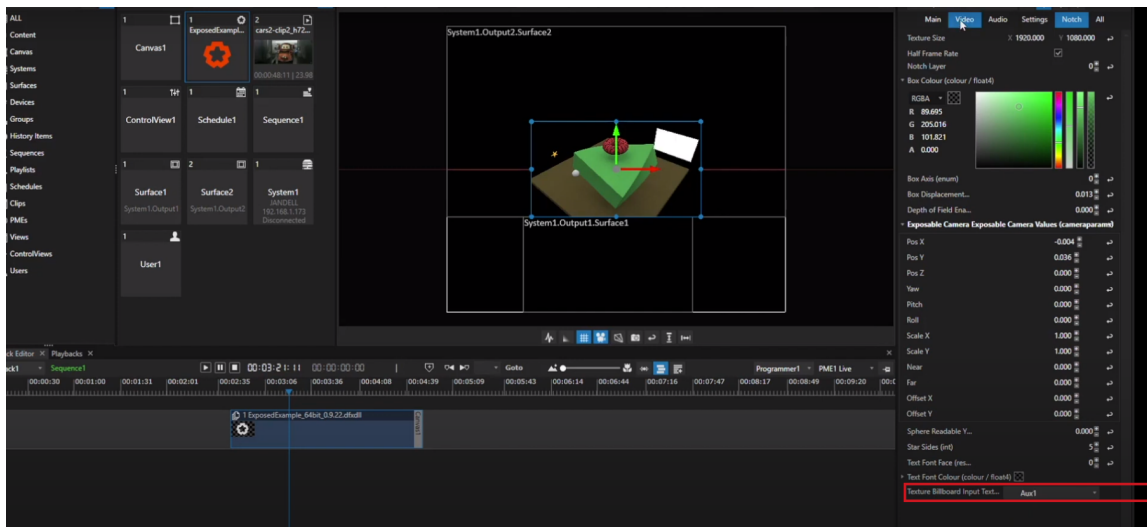
## Keyframes

- All parameters that are listed into the Inspector for the selected Notch Block are also available as Keyframes for animation
- Double-Click with your mouse on the Clip Container to open the [Keyframe-Editor](#) and choose your property for animation from the drop down list

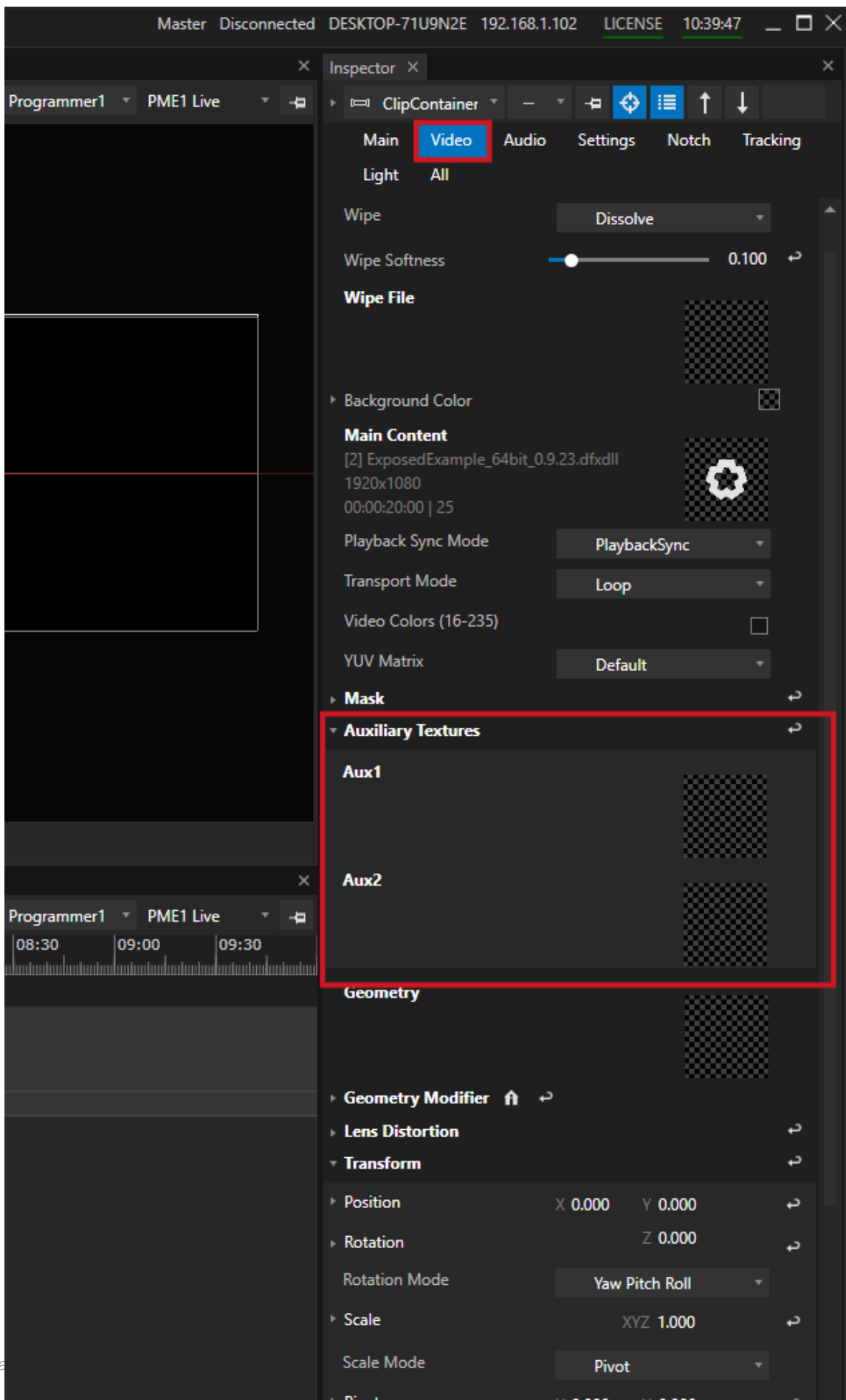


## Auxiliary textures

- When your Notch Block includes textures that could be assigned to an element, there is an option to work with auxiliary textures that are provided from another content element into VERTEX

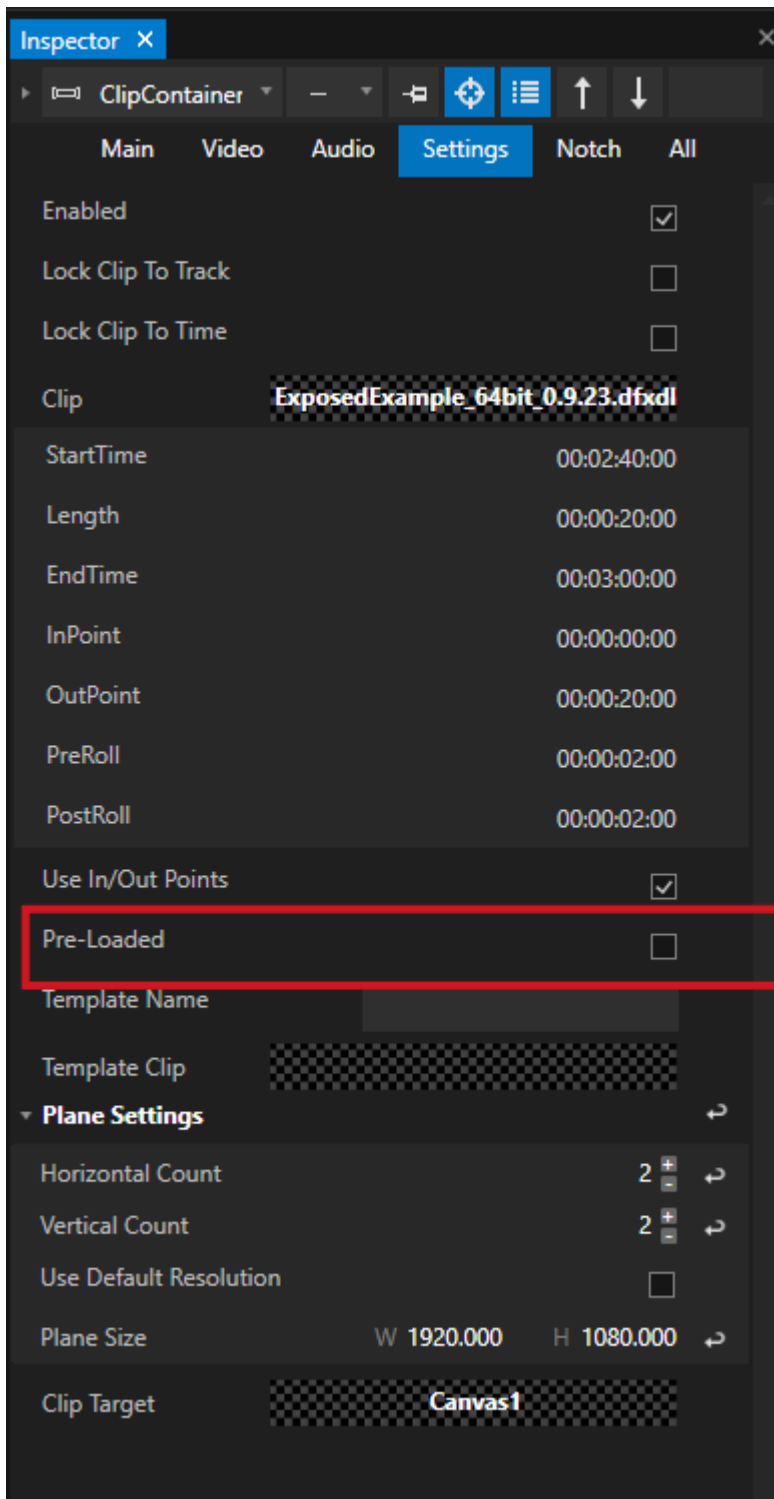


- Just choose e.g. AUX1 in the texture dropdown list
- then switch to the video tab and change [Inspector Mode to Advanced](#)
- Drag and drop a Video, [Live Input](#) or another content asset from the [Project Explorer](#) to the Auxiliary Textures field



## Proload Notch Blocks

- *To minimize loading time and calculation times it also possible to preload a Clip Container with a Notch Block*
- *When enabled, Clip Containers are pre calculated and pre loaded into the GPU cache. The content directly is available and rendered when the Playhead runs into the Clip Container.*
- *Pre-loaded Content claims hardware resources of your GPU- please use carefully and monitor the hardware usage!*





#### **System Performance and System Load**

The Systems performance depends on your hardware setup and the settings made in Notch  
For Notch performance optimization and measurement, please read the documentation from  
Notch about [Managing Performance in Media Servers](#)

VERTEX provides a setting to halve the frame rate for Notch Content which reduces the local load on your  
VERTEX System (e.g. for testings and programming)

## Further Information



[Notch website and documentation](#)

### 5.4.9 PSD-Files

- VERTEX is able to read and import Files from Adobe Photoshop including layers
- Layers has to be rasterized in Photoshop before.
- Each single Layer is accessible as child element of the main content item and can individually be placed into Clip Containers or as Clip into a Playlist

### 5.4.10 Live Input

- VERTEX supports **different types of Live Inputs and Capture Cards**
- Live Inputs that are available for a System are **shown in the Project Explorer as child elements of this System**
- **Live Inputs first have to be added as Content** to your Project (Right-Click on Live-Input -> Add to project)

# Capture Cards

## SUPPORTED MODELS AND MANUFACTURERS

VERTEX is shipped with a support of video input cards of the following manufacturers:

- AJA
- Blackmagic Design
- Bluefish444
- Magewell
- Datapath
- DeltaCast
- Deltacast FLEX
- Magewell
- Osprey
- Streamlabs

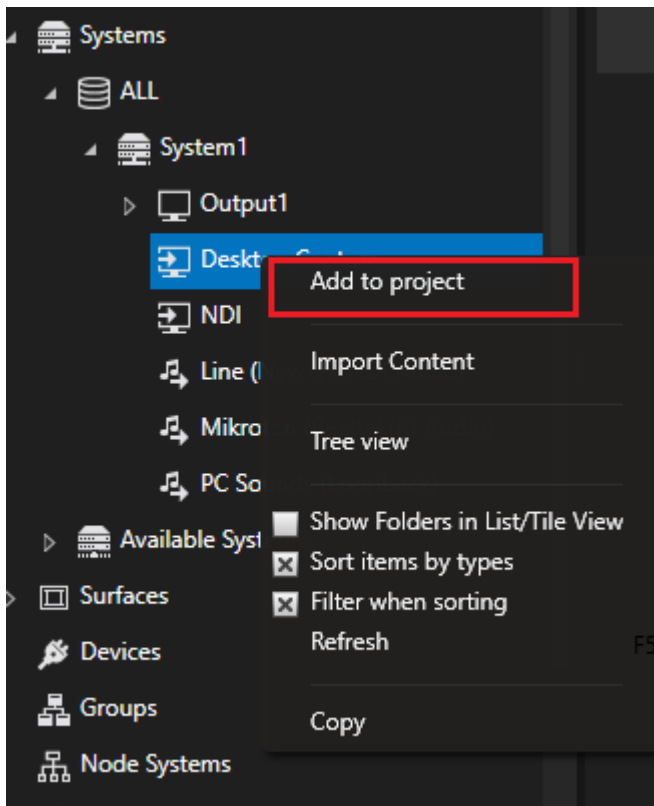


Follow the instructions of the manufacturer installing the hardware and the driver.  
Once the driver is installed, VERTEX should detect the input card in the project explorer as child element of your system.

### **Questions about a certain capture card or model?**

First use our free VERTEX trial to test your hardware. for questions write an email to [support@ioversal.com](mailto:support@ioversal.com)

## ADD LIVE INPUT TO YOUR PROJECT



After an Input Card and its driver is installed, the Input Card should be listed as child element of your Vertex System into Project Explorer.

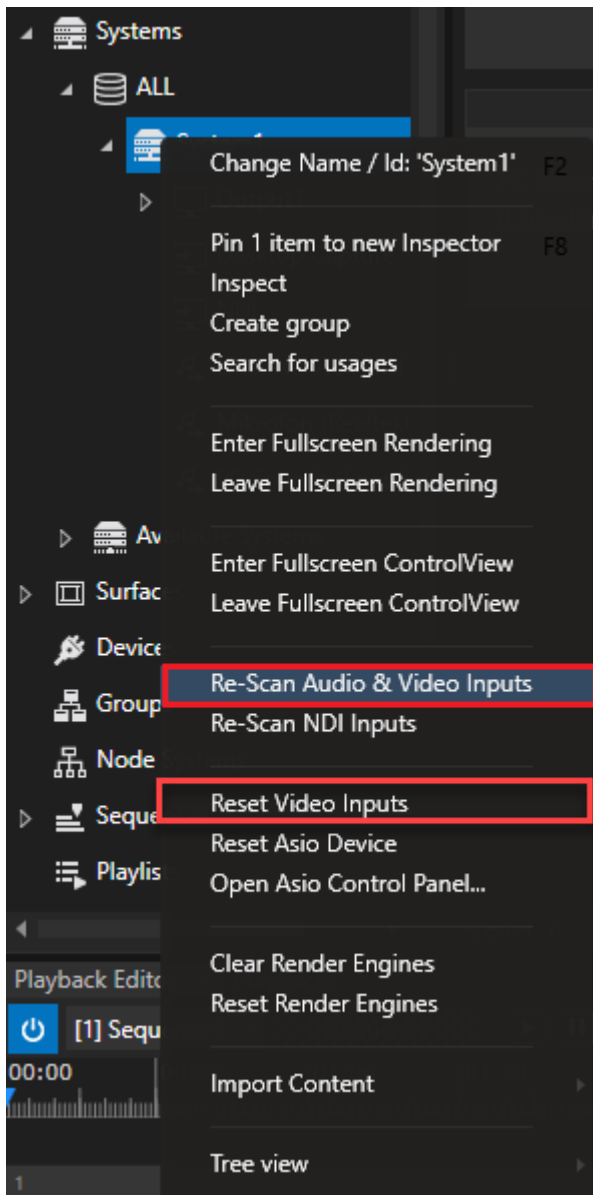
- go to **project explorer** and navigate to the **system** where the input device is installed
- **select the input device**
- right-click with your mouse and **open the context menu**
- select **"Add to project"**
- the input device now is listed as a **content item** in the project explorer
- **drag** the live input content into the playback editor

## RE-SCAN VIDEO SOURCES

If your **input device is not listed as a child element** of a system in the project explorer, please refresh the video sources:

- select **System** in project explorer
- right-click with your mouse and open context menu
- select **"Re-Scan Audio & Video Inputs"**





If the **input texture** of your capture card has a **wrong resolution**, try to reset your video inputs

- select **System** in project explorer
- right-click with your mouse and open the context menu
- select "**Reset Video Inputs**"

## NDI

With a built in NDI receiver, VERTEX is able to receive NewTek™ NDI streams.

- **VERTEX automatically detects NDI streams** into your network.
- Each detected stream **is listed as a Live Input Element** as child element of your VERTEX System into Project Explorer

When no NDI Stream or a wrong number of streams is detected, try to refresh the **Video Sources of your System**:

- Select your System into Project Explorer
- Right-click with your mouse to open the Context Menu
- Click to "Refresh Video Sources"



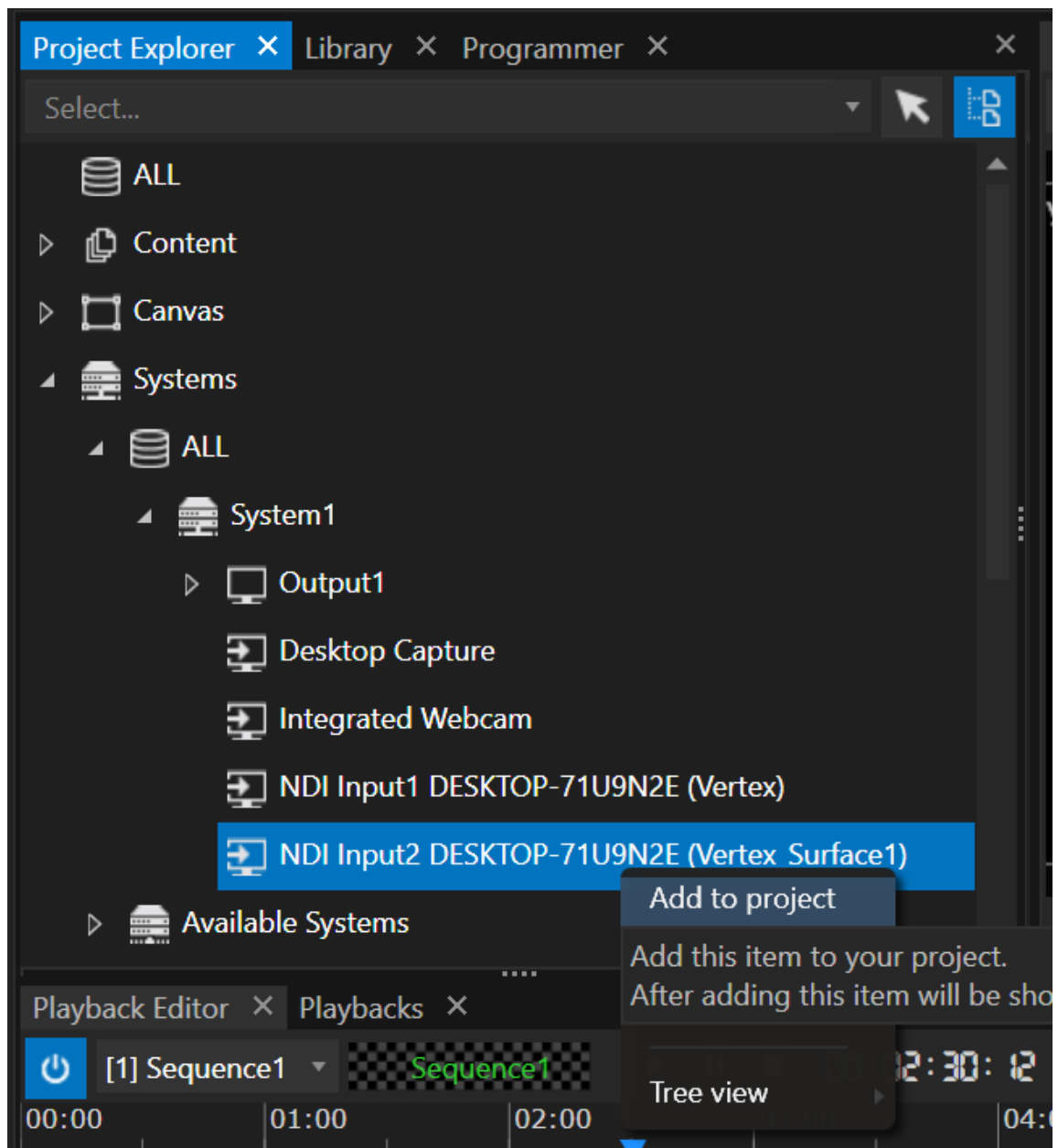
**If NewTek™ NDI Tools is installed on your PC:**

If the NewTek™ NDI Tools are installed on your PC with in parallel installed VERTEX, there will be an additional NDI Input Source called NewTek NDI Video.

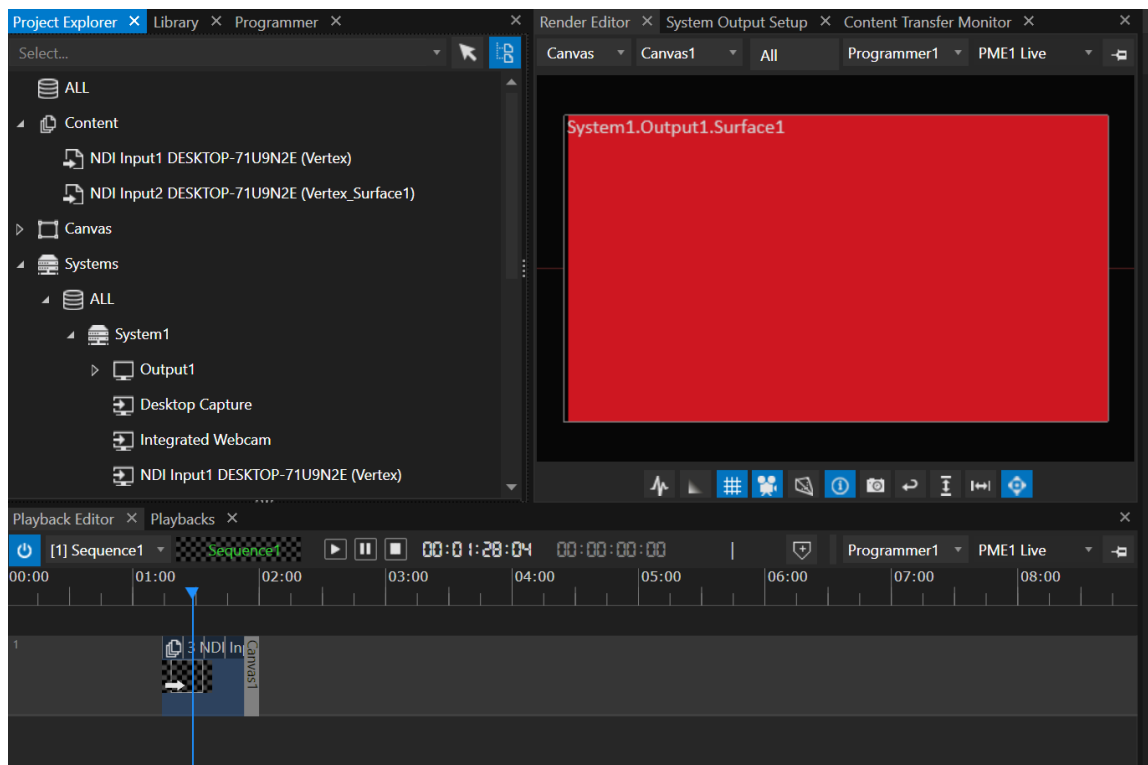
We recommend to use the VERTEX NDI Receiver. This kind of input source is independent from an installation of the NewTek™ NDI Tools on any other PC or Client of your project. The VERTEX NDI Receiver will be available on any Hardware on which VERTEX is installed

## Add NDI to your Project

- Navigate to your **System into Project Explorer**
- Open the tree and **show child elements of your System**
- **Right-Click with your mouse** onto NDI Receiver



- **"Add to Project"**
- The NDI Receiver now is listed in the **Content Section** into Project Explorer
- **Drag this NDI Content to the timeline of the Playback Editor** and create a Clip Container



## Desktop Capture

- Captures the Windows Desktop of a System
- Shows Mouse or Touch Interactions

### Add a Desktop Capture to your Project

- Navigate to your System into Project Explorer
- Open the tree and show child elements of your System
- Right-Click with your mouse onto Desktop Capture
- "Add to Project"
- The Desktop Capture now is listed in the Content Section into Project Explorer
- Drag this Desktop Capture Content the timeline of the Playback Editor to create a Clip Container

## Camera

VERTEX supports all cameras that are working with Microsoft Windows after drivers are installed:

- *integrated webcams*
- *USB webcams*
- *some ethernet-based cameras*



Note: at the start of VERTEX and when loading a project, VERTEX initializes all input devices. A connected webcam's LED may flash briefly, but VERTEX is not recording anything.

## Add Camera Live-Input to your Project

- *Go to Project Explorer and navigate to your local system*
- *If cameras are detected from VERTEX, they should be listed as child elements of this system.*
- *Open the context menu with a right-click.*
- *"Add to project"*
- *The camera is now listed as a content item in the content section.*
- *Drag the camera content as a clip container into the timeline of the Playback Editor.*

## Spout



Spout is an open source video texture sharing framework for Microsoft Windows. With Spout it is possible to share video or graphics textures in realtime between windows applications on the same System.

Spout is supported and integrated by a lot of creative applications.

All information including a Spout test sender and receiver could be downloaded on the

[Spout Website](#)

## Add Spout to your project

- Check if there is an active spout sender from another application active on your Windows System
- Go to the VERTEX Project Explorer and navigate to your local System there
- Spout should now be listed as Live Input and child element of your local System
- Right-Click on the Spout element and select "Add to project"
- Into the Content Section, Spout now is listed as Content of your project
- Select Spout from Content and drag it to the Playback Editors timeline to create a clip container



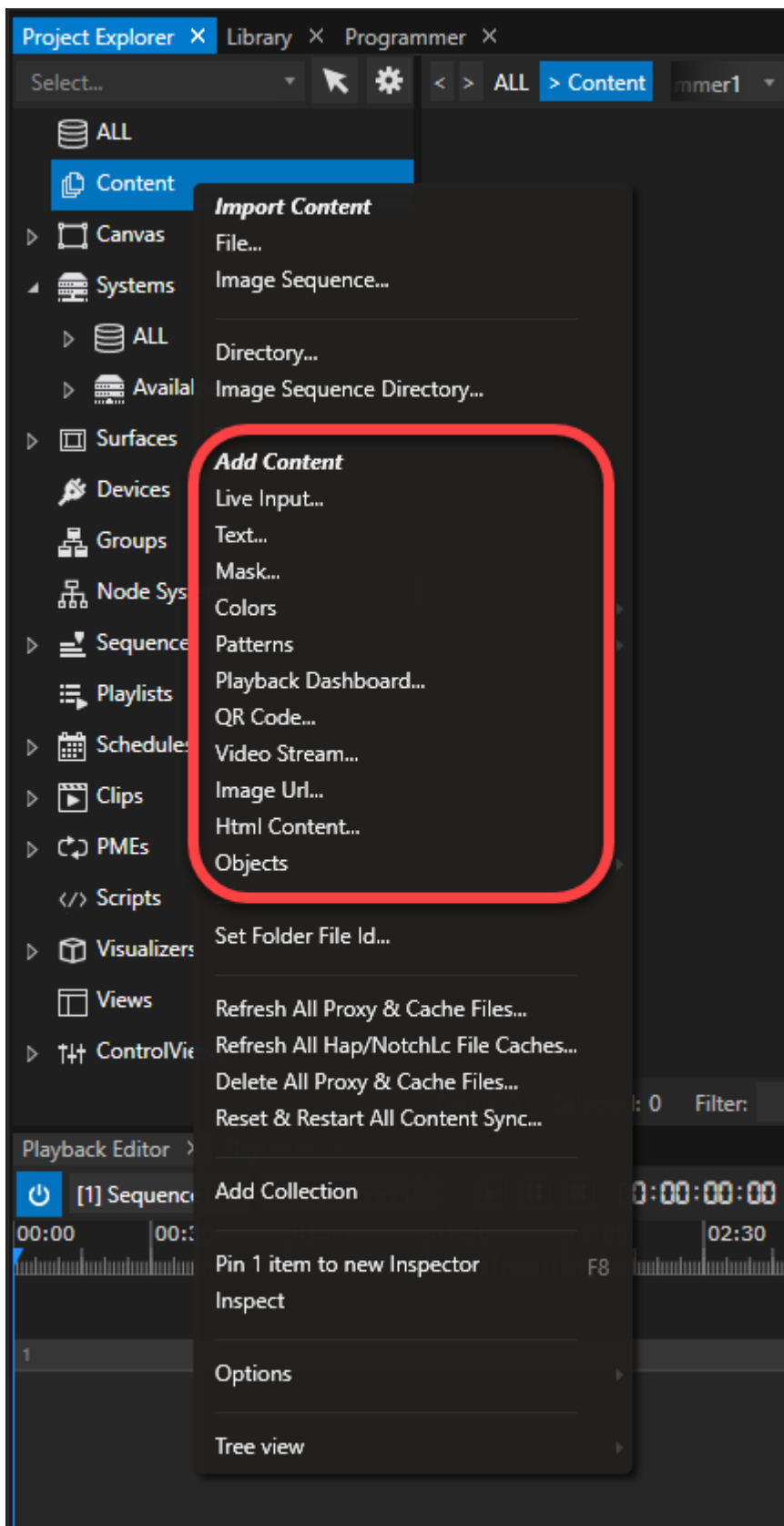
If Spout is not listed as child element of your local System: right-click with your mouse on the System, and select "Refresh Live Inputs" in the Context Menu.

If no Spout Sender is detected, no child entry is shown!

## 5.4.11 Generative Patterns

- Generative content is **static content** that is generated **based on your settings**.
- You have access to **solid color, gradients** or **test- or UV-patterns** directly from VERTEX, no needs to generate them with an external application.
- Once added as content to your project, you are able to **change and adapt the settings in the inspector**. Based on these settings, the visual result changes.

## Add a Generative Pattern





1. go to *Project Explorer*
2. right-click with your mouse to the *Content* section to open the context menu

or

1. go to the main menu at the top bar *Create -> Content*



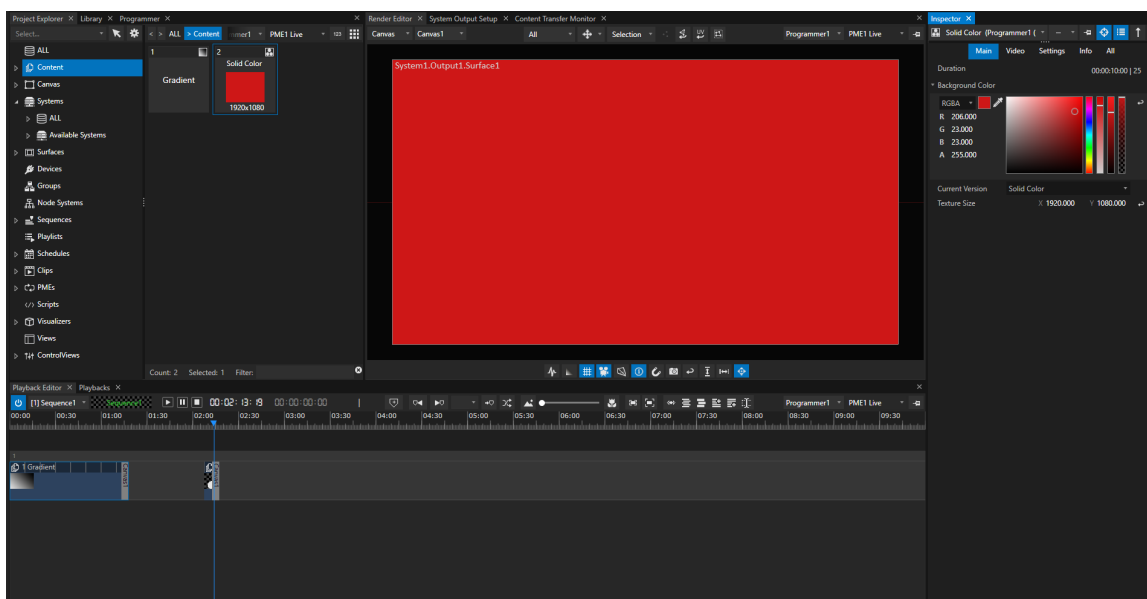
The render texture size defines the size of each of the content items below.  
By default this size is set to 1920x1080 pixel. You can change the size in the [inspector](#)

## Solid Color

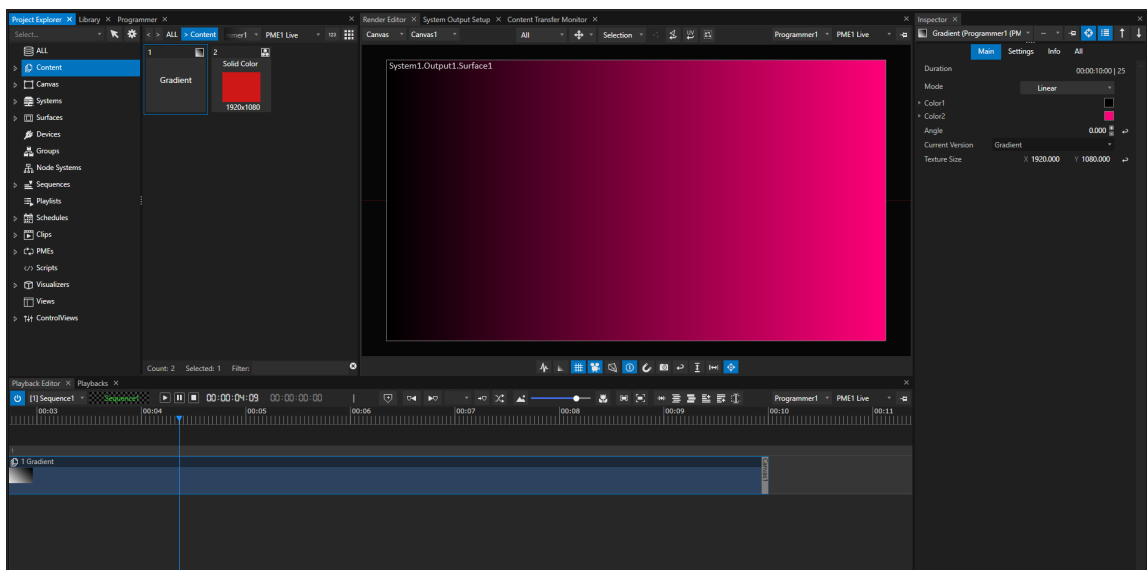
- solid color is a content element with a single solid color texture
- select the solid color element in the project explorer to set color and texture size



After applying changes to this content item, you might have to refresh the clip container by sliding the playhead in and out of the clip container in order for the changes to become effective.

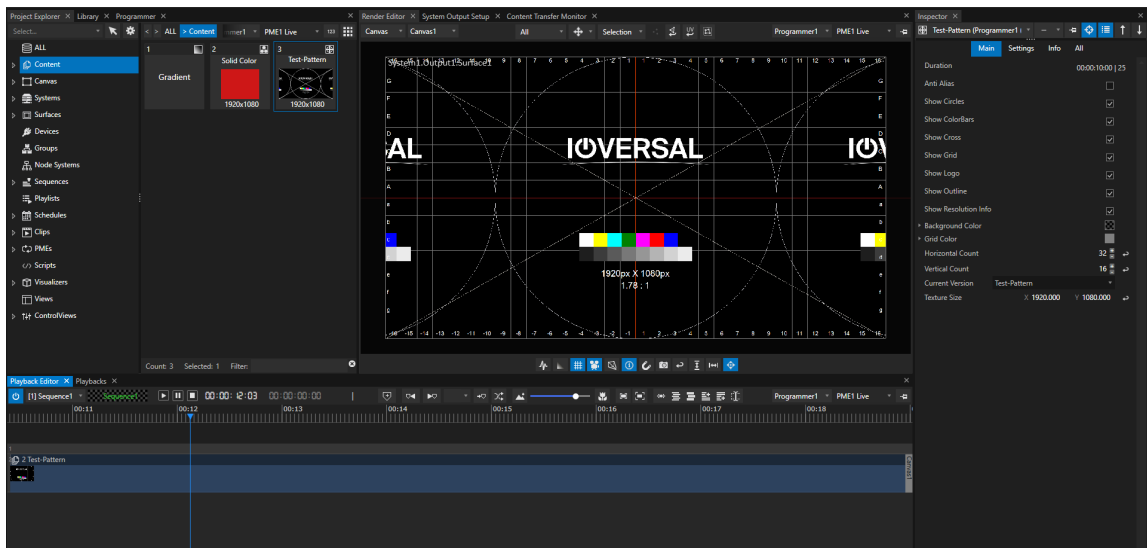


## Gradient



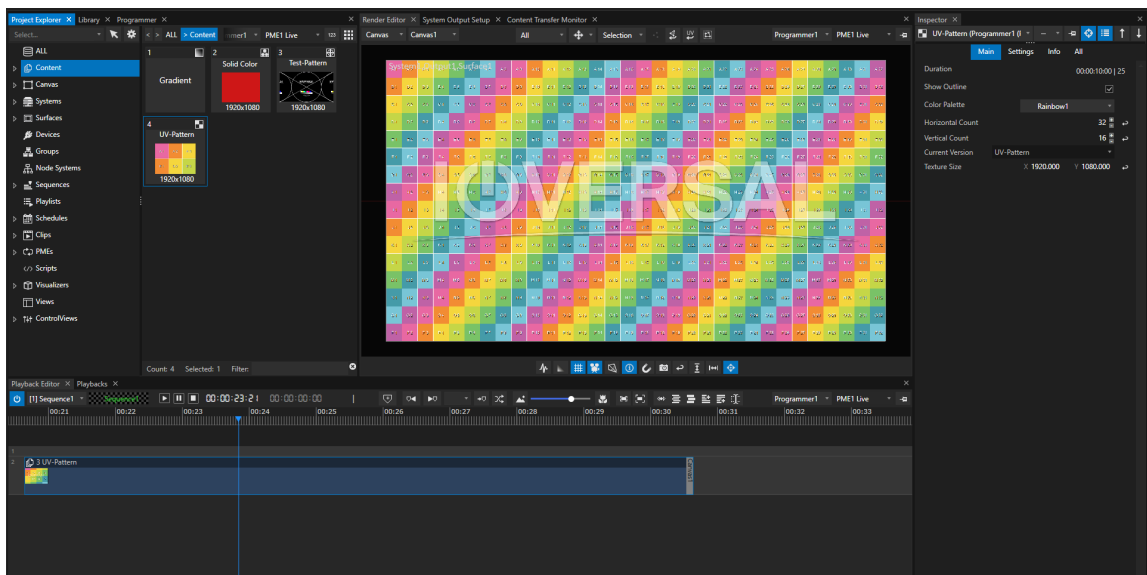
## Test-Pattern

- Based on the texture size you set in the inspector, VERTEX calculates and draws a test pattern.
- All parameters are based on the texture size and the horizontal and vertical count of elements.



## UV-Pattern

- Based on the texture size you set in the inspector, VERTEX calculates and draws a UV-Test-Pattern.
- All parameters are calculated based on the texture size.

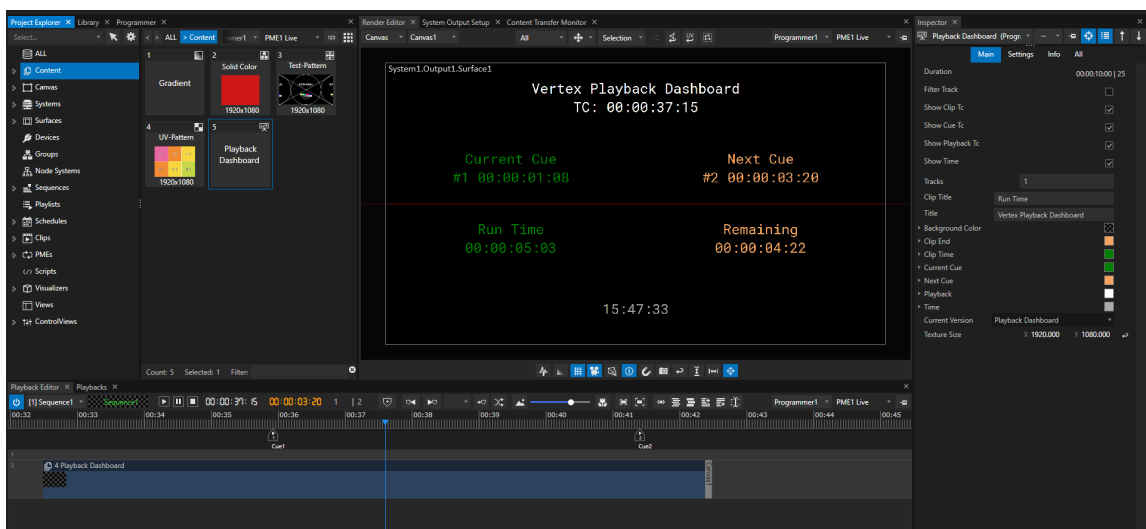




After applying changes to this content item, you might have to refresh the clip container by sliding the playhead in and out of the clip container in order for the changes to become effective.

## Playback Dashboard

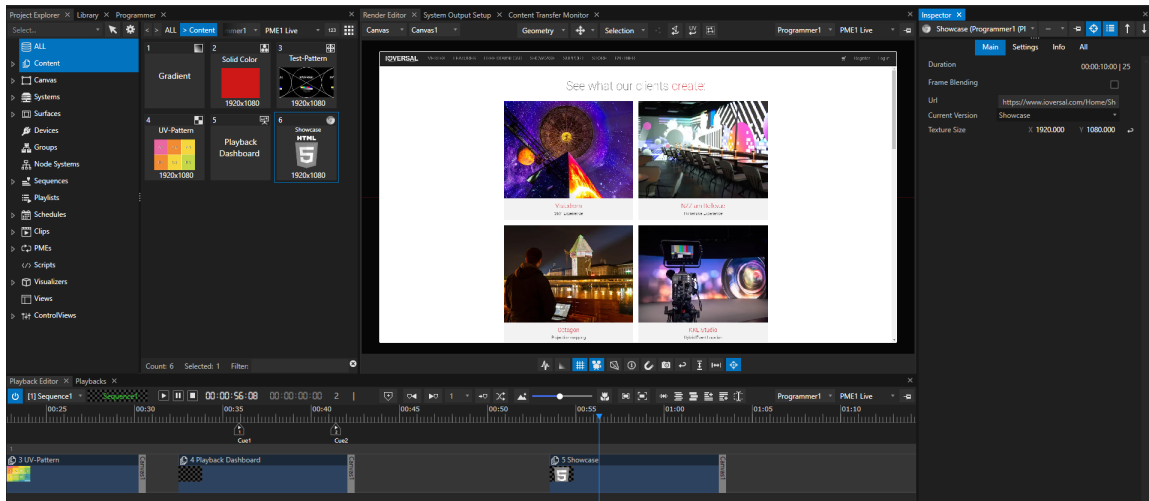
- Based on the texture size you set in the inspector, VERTEX calculates and draws a playback dashboard.
- All parameters and information are based on the timecode and the cue positions of your playback.



After applying changes to this content item, you might have to refresh the clip container by sliding the playhead in and out of the clip container in order for the changes to become effective.

### 5.4.12 HTML-Content

- VERTEX offers a **HTML 5 render engine** - a browser based on the chromium code
- VERTEX HTML content **supports all common browser functionality** including JavaScript - for tasks like CSS Animations
- Works with both **local HTML pages** or **internet/intranet URL**

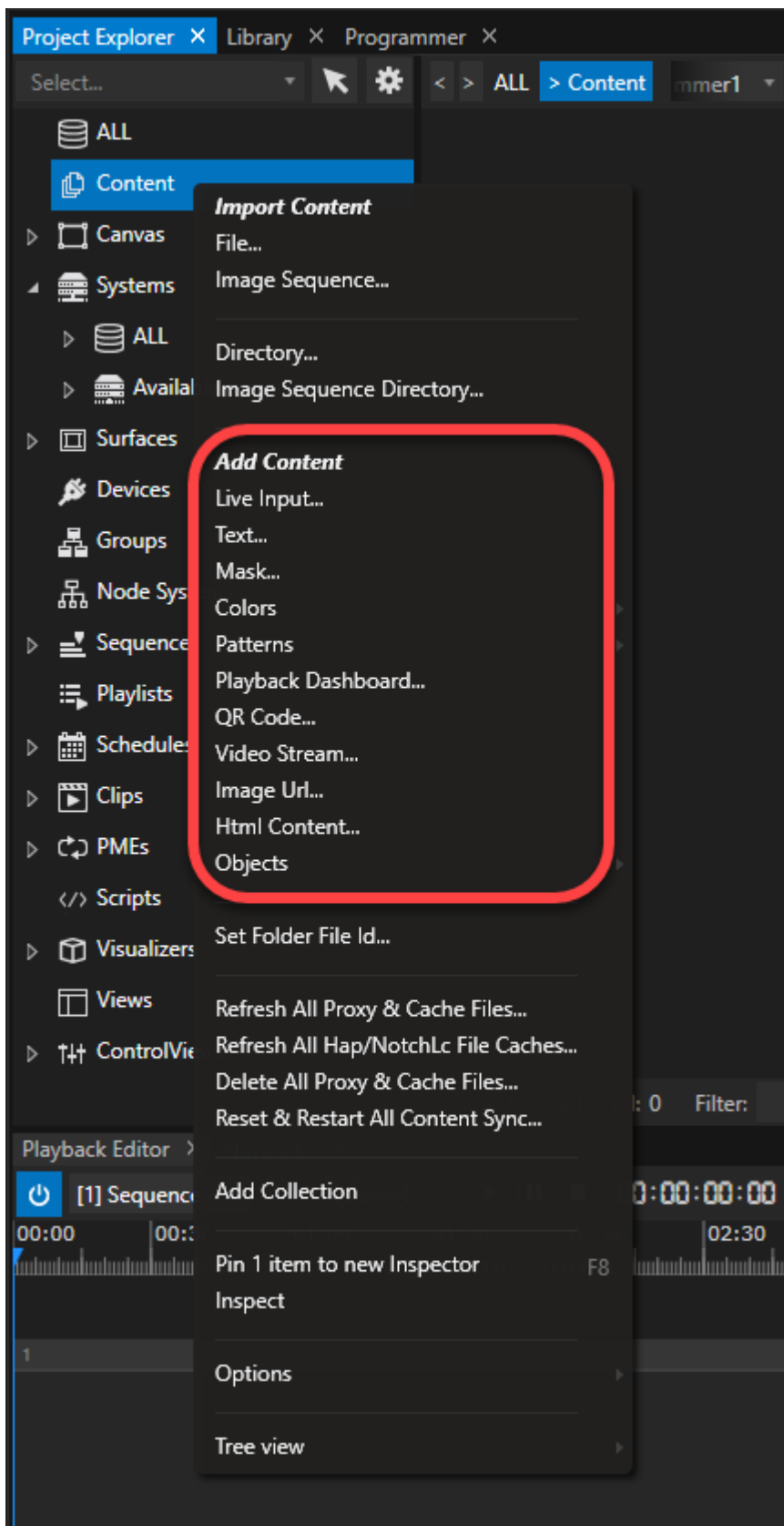


## Add HTML Content to your Project

1. Go to project explorer
2. Right-click in the content section to open context menu
3. select **Add...HTML Content**

or

Go to the main menu at the top bar and select **Create -> Content -> HTML Content**



## Settings

### URL

- can be an internet or intranet URL
- could also be a file path to a local HTML file

Make sure

- that the systems that render URL content have internet/intranet access.
- the system's [fullscreen interaction settings](#) enable the use of mouse or keyboard for full web browser functionality.



Opacity/Transparency for HTML 5 is supported by the VERTEX HTML Browser Engine

### Render Texture Size

- The render texture size defines the size of your HTML browser item.
- By default this size is set to 1920x1080 pixel. You can change the size in the [inspector](#).
- If using responsive web content that is optimized for tablets or mobile phones, the view mode of the HTML content is adapted responsive for small render texture sizes.

### FPS

- Frames per Second (FPS) defines how often the browser engine renders and refreshes your HTML content. Please keep in mind, that this number is related to the number of frames the browser engine renders the HTML content and not on URL reloading.



To save your system's resources, dial down the FPS on static HTML pages without CSS Animations to 1.

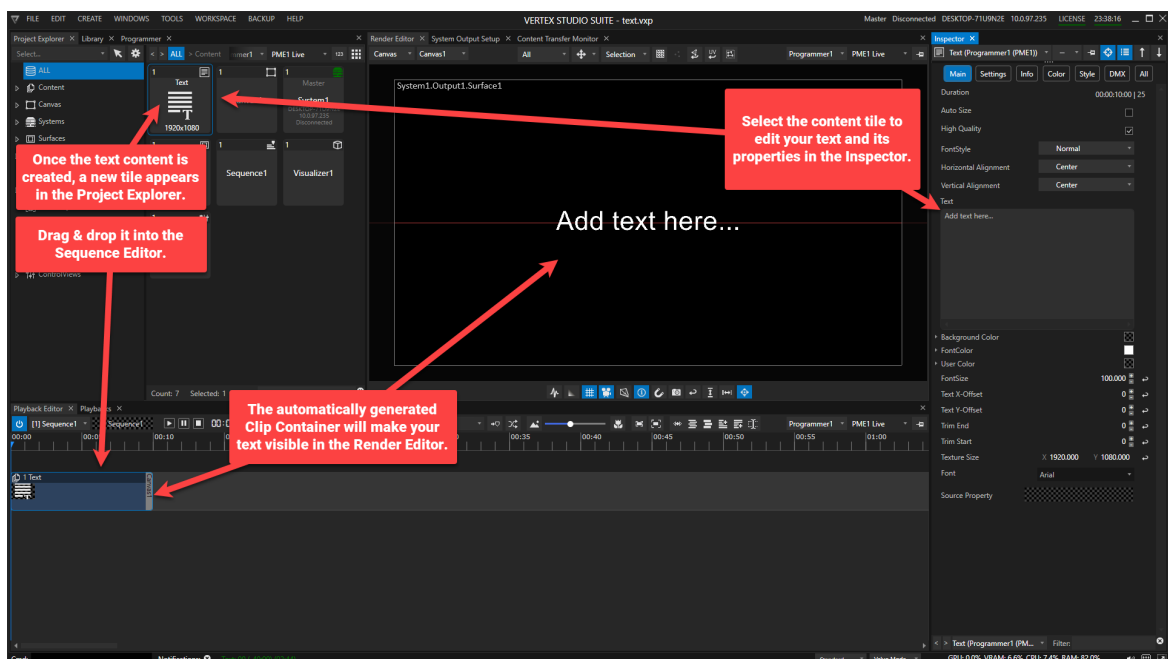
Each HTML content in VERTEX opens its own browser engine. When working with different HTML content elements or big render texture sizes, a lower FPS number for HTML rendering can save processing power of your render editor or the fullscreen renderer.

## 5.4.13 Text

- This feature **generates a text based content**.
- **Font, size, color and background** can be adjusted in the inspector.
- All fonts installed under Windows are supported.

## Working with Text Content

- Text content can be added via context menu by right-click on **CONTENT** in the Project Explorer.
- Alternatively, go to: **MAIN MENU > CREATE > CONTENT > TEXT**



- The Inspector setting for Source Property can be used to display any value VERTEX processes. If you wish to incorporate any values in addition to your text, please type **{0}** into the text field.



**Texture Size**

*As for all other content items in VERTEX, the texture size defines the pixel size of your text element.*

*By default this size is set to 1920 x1080 px but can be adjusted to your needs.*

## 5.5 Create your first project

- We want that your start with VERTEX is **as easy as possible**. As Beginner or as Expert.
- When creating a **new project VERTEX handles some settings in the background to give you a comfortable start**.
- Both Workspace and Output Settings are **preconfigured based on your Windows System settings**.

### Steps

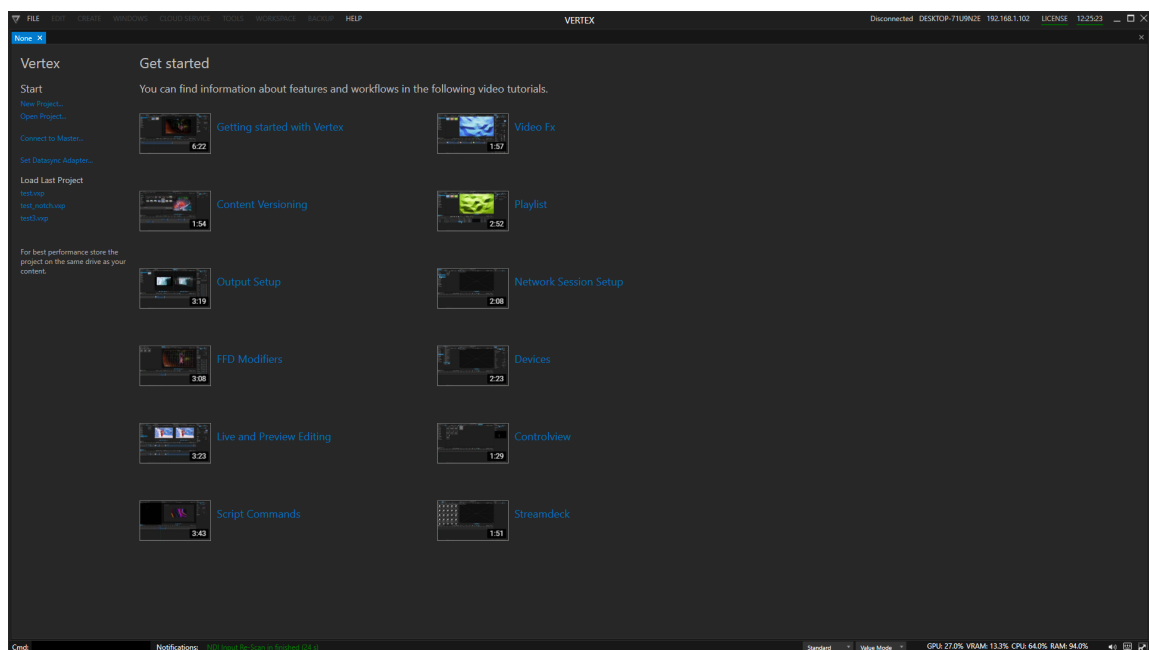
#### 1. Install VERTEX

Need help? Here is a step by step guide with all information: [Installation](#)

#### 2. Start VERTEX

VERTEX starts with a splash screen that is shown for a short time

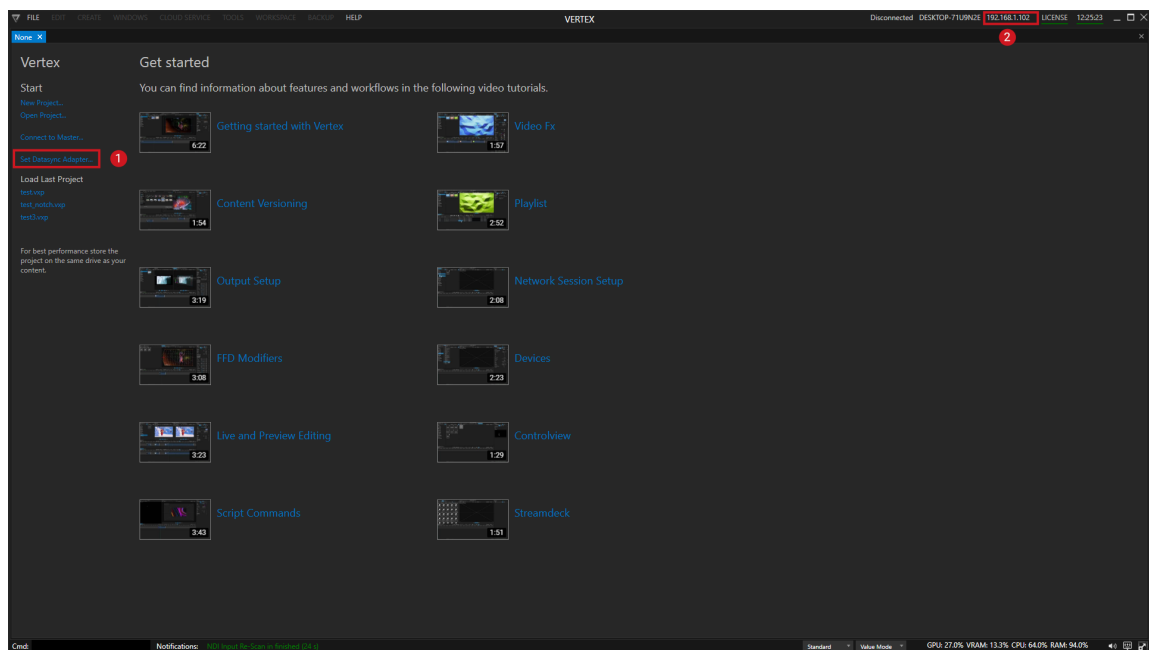
The UI is loaded and you will see the Startup Page first



### 3. Check/Set Data Sync Adapter

When you want to use your VERTEX System in a network [session with other VERTEX Systems](#), please double check for the right Data Sync Adapter.

This can avoid hassle later on when setting up a session.



<b>1</b>	Select "Set Datasync Adapter" to initially set your preferred network adapter. If Ethernet is available, we strictly recommend to used a fast and cabled network instead of a wireless one
<b>2</b>	The Data Sync IP is shown at the Top bar. Please check if the correct IP address is displayed there. If no IP is displayed, please set Data sync Adapter again

### 4. Create a new VERTEX project

**Project Path**

To run VERTEX with the best playback performance, we strictly recommend to store your VERTEX project folder on the same drive as your content is located.

Please use a fast drive for both to ensure an optimal performance of the software.

## 5. Wait until the project is created and ready to work on

The project will be loaded and ready to go for your first experience with VERTEX-

You will see:

- o a **preconfigured Workspace**

with the most important tools and windows

- o a **preconfigured Canvas and Output Configuration**

The Canvas size matches with the Windows 10 screen settings of your local System. Canvas 1 is preconfigured with the total pixels of the Windows 10 desktop you have set. For each the outputs that are already connected to your PC a Surface is created and already assigned to your Canvas.

**Setup Example**

2 Full-HD Screens are connected to your hardware System. Both screens are arranged side by side into your GPU driver/the Windows 10 desktop settings. Your initial Canvas size is  $1920 \times 2 = 3840$  pixel width and 1080 pixels in height.

For each the Screens that are connected to one of your GPU outputs, VERTEX creates a Surface with 1920x1080px. The initial arrangement of this Surface into your VERTEX Canvas should match with the settings you have made for your Windows 10 desktop.

## 6. Explore and learn VERTEX

**Learn** the VERTEX basics or **explore** the software by yourself:

- Check out the [ioversal Youtube channel](#) - you will find there some **quick basic tutorials**
- Read this manual** and **follow the next steps**:
  - [Become familiar with the user interface](#)
  - [Import Content](#)
  - [Manage Content](#)
  - [Arranging Content](#)
  - [Set Outputs](#)

- [Playback Live and/or in Preview](#)

## 5.5.1 Manage Content

- VERTEX offers you **different tools** to support you in **managing and structure** your project assets
- **User Properties** help you to organize each single element into VERTEX **with meta data** and specific color
- With **Collection and Groups**, you are able to structure content and items into the Project Explorer
- VERTEX comes with an easy-going **versioning** feature: Just import a new version and **change the content everywhere into your playbacks**
- **Automate Content Import and Playlist/Sequence creation with WatchFolders**
- Set **Target Systems** to transfer a content only to a certain System

## Metadata: User Properties

Set custom User Properties for each item: Content, Clip Containers and Devices. Add personal notes to remember

- [Notes](#)
- [ID](#)
- [Name](#)
- [User Color](#)
- [Tags](#)

## Project Explorer

Order Content, Devices and all other items into Project Explorer. Create your own project structure by using Collections as folders for your favorite items:

- [Collections](#)
- [Groups](#)

## Content Versioning

Import new versions of your content and easily switch between versions: The new content version is automatically replaced on all places into your projects where it is used.

Check the Topic below for more details:

- [Versioning](#)

## WatchFolder

Observe Windows folders. VERTEX automatically imports new files to your project. You are able to let VERTEX create Playlists or Sequences based on this content

- [WatchFolder](#)

## Advanced: Target Systems for Content Sharing

Set targets where your content is transferred to. Only share content data to a certain System into your project  
Create target "templates" with use of Collections.

Settings for Target Systems only become relevant if you work with [several VERTEX systems in one project](#) and want to distribute some content to different systems.

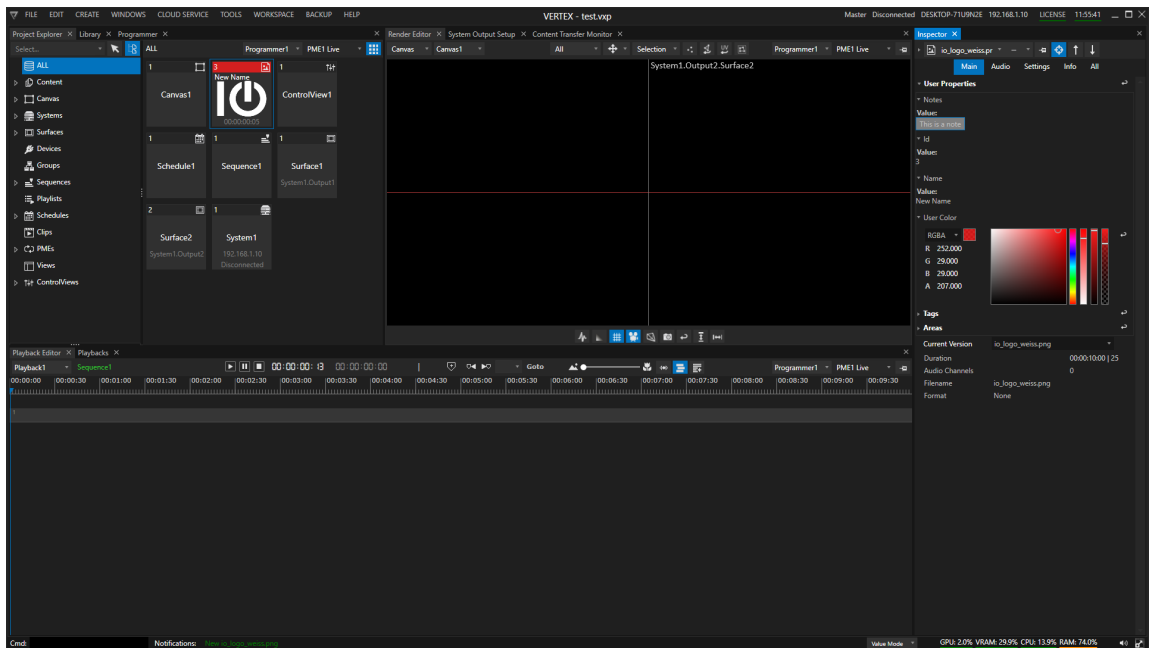
- [Target Systems](#)

### 5.5.1.1 User Properties

- **For each item** in VERTEX, individual properties for content management can be set
- This User Properties **can be set in the Inspector**

## Notes

- You can add notes for **every item** of your VERTEX Project
- Notes are shown as **child elements of the "User Properties" into [Inspector](#)**
- Just **double click into the value** field to enter a new value



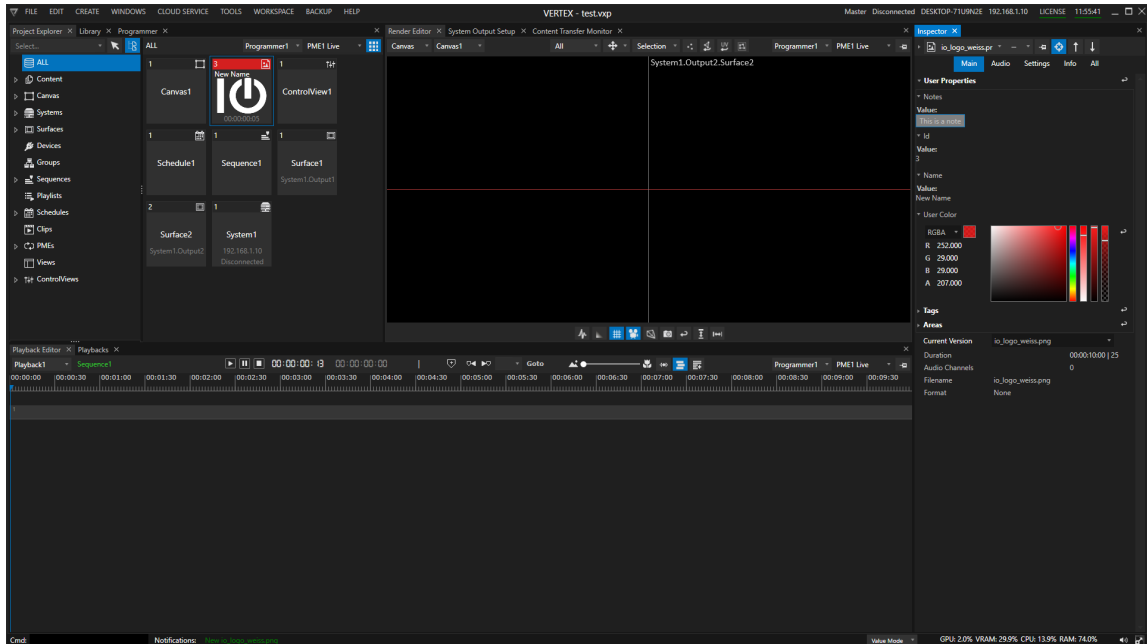
## ID

- The ID of a content is displayed into Project Explorer as number for each item
- For Clip Containers, the ID is displayed as number for each Container into playback editor
- be careful when changing this ID!

## Name

- Enter a custom name for your project item

- When rename an item into Project Explorer (via context menu or via Shortcut "F2") this name also is set as name property into User Properties

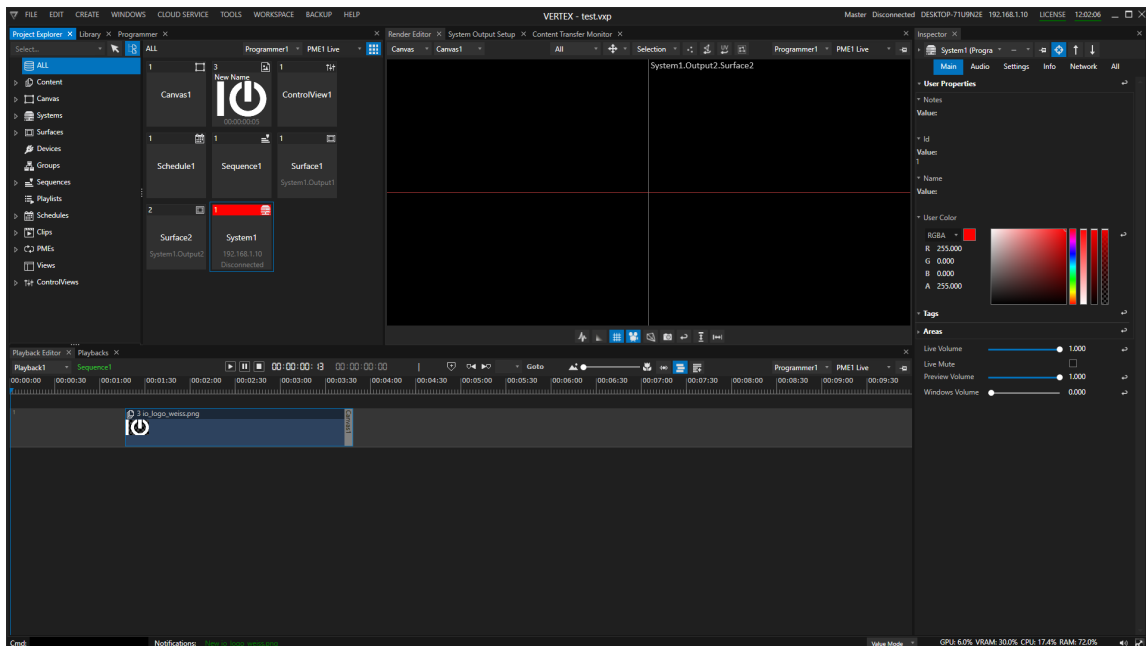


Enter a custom name for your project or content item.  
The current Vertex version shows this custom name for most of the items also into Project Explorer. Please note: For some project items this is not implemented yet.

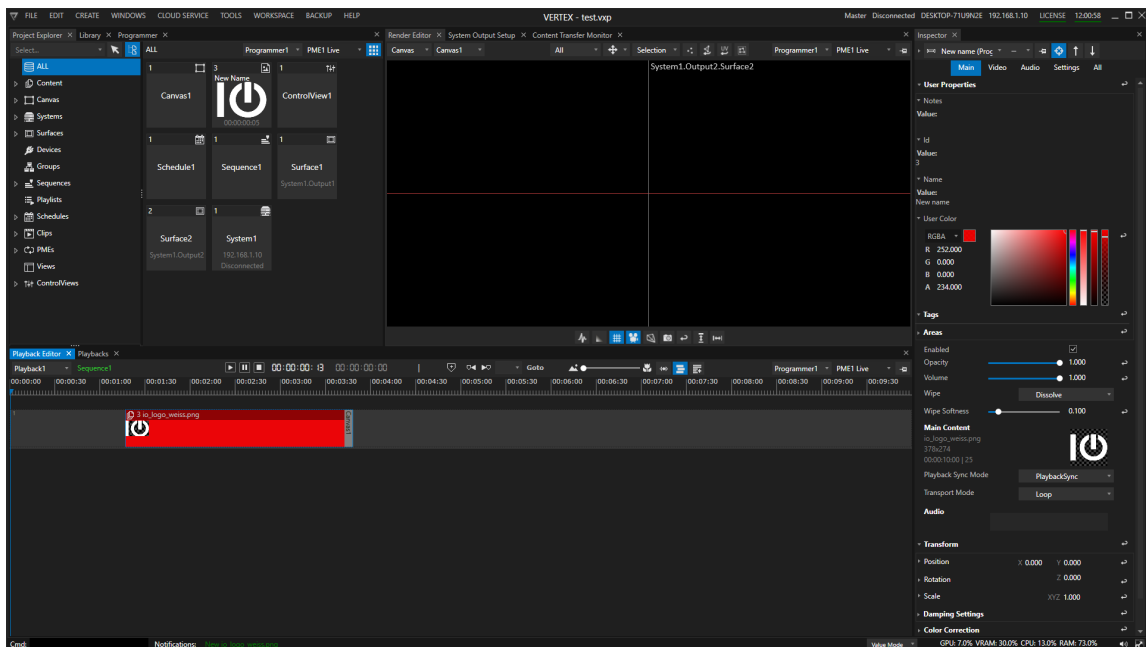
## User Color

- Use User Colors to **increase the visual overview** in your project or to make **thematic clusters**
- User Colors for Items **are shown into Project Explorer**
- User Colors for Clip Containers are also shown into Playback editor





Settings could be done into inspector - User Colors are child Elements of the User Properties



Set a User Color for a Clip Container

## Tags

- Assign Tags to project items to order them by subject

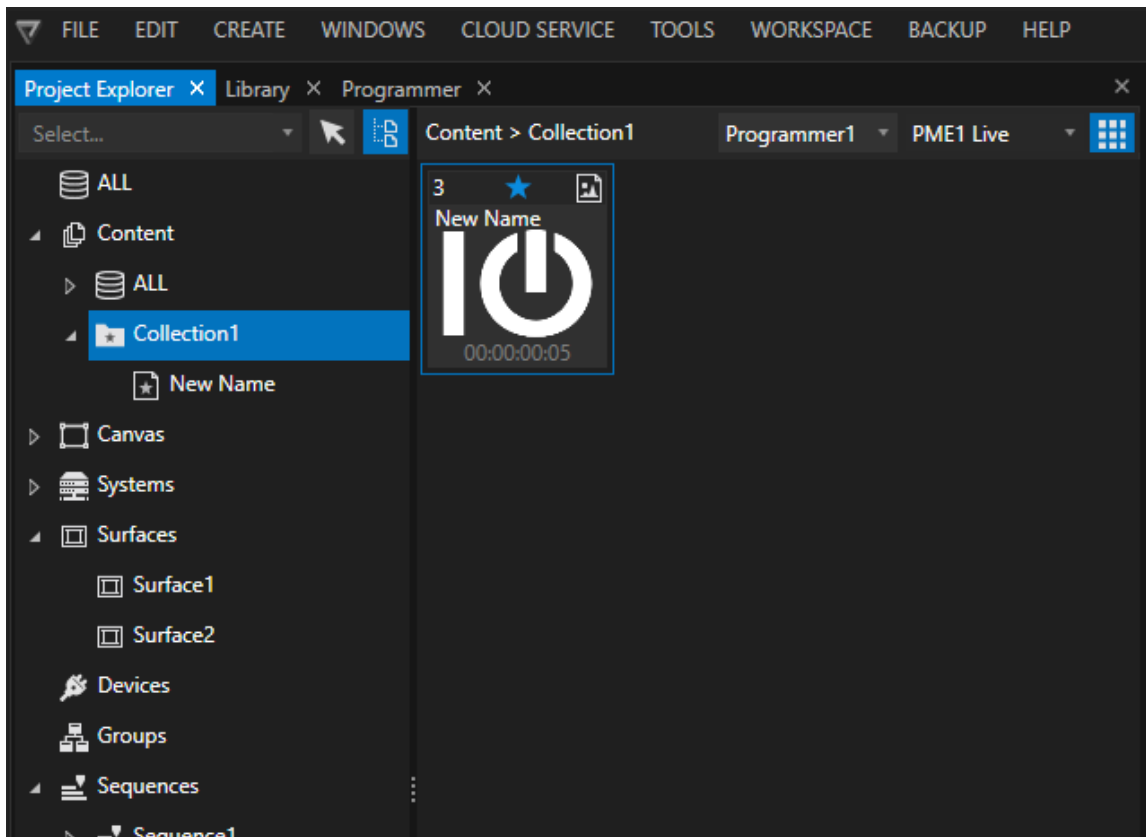


**Feature is prepared but will come in a future VERTEX release**

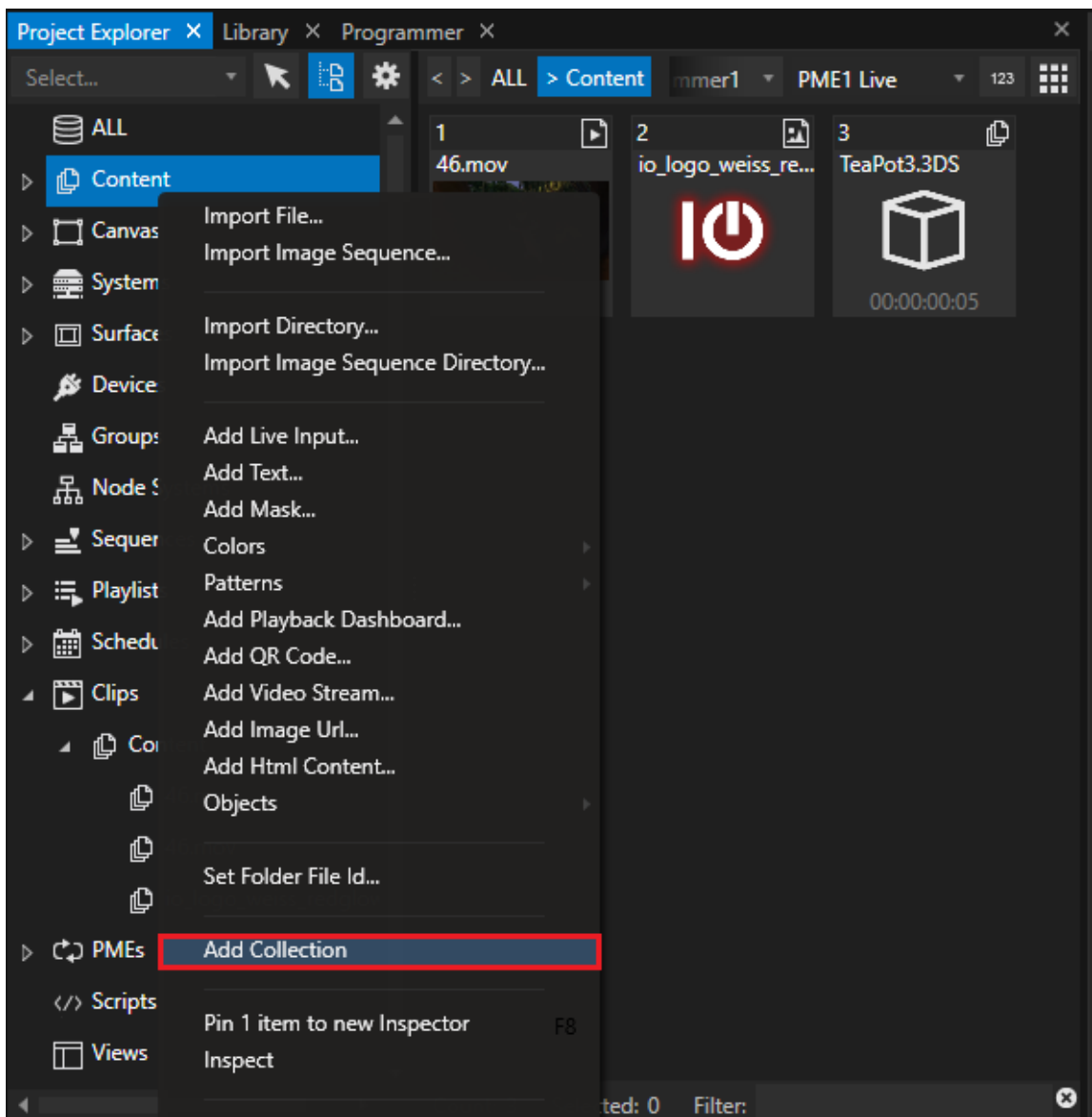
*This functionality will come with future VERTEX releases and is not fully implemented yet.*

### 5.5.1.2 Collection

- Collections are your **virtual folders**, where content and items can be sorted into and **"Bookmarked"**
- Collections **can be created for every category** into the **Project Explorer**
- When you **delete an item from a favorite folder**, it is **still available into the "All" category** - only the bookmark is deleted



## Create a Collection



1. Select with your Mouse a Category into Project Explorer
  2. Rightclick and open Context Menu
  3. "Add Collection"
  4. When a Collection is created into Project Explorer, VERTEX additionally introduces an "All" sub category where all items of your parent Category are sorted in automatically.
  5. Select with your mouse an item from this "All" Category and drag it to your Collection "
  6. Select the new Collection with your Mouse and enter a custom name with help of the context menu or the shortcut "F2"
- You are able to create a **Collection as a child of another Collection**.

- Create a Collection is possible **for every Category** into Project Explorer, e.g. also for Systems, Canvas, Surfaces



When a Collection is created into Project Explorer, VERTEX additionally introduces an "All" sub category where all items of your parent Category are sorted in automatically.

## Advanced: Content Sharing to certain Target Systems with help of Collections

- With the help of Collections you are able to create advanced content sharing presets
- Collections that are created into the "All" Category into Project Explorer can host items from different Categories -e.g. Systems and also Content
- When 1 or multiple Systems were added into this kind of Collection, all Content that is imported into this category, only is shared to this Target Systems.

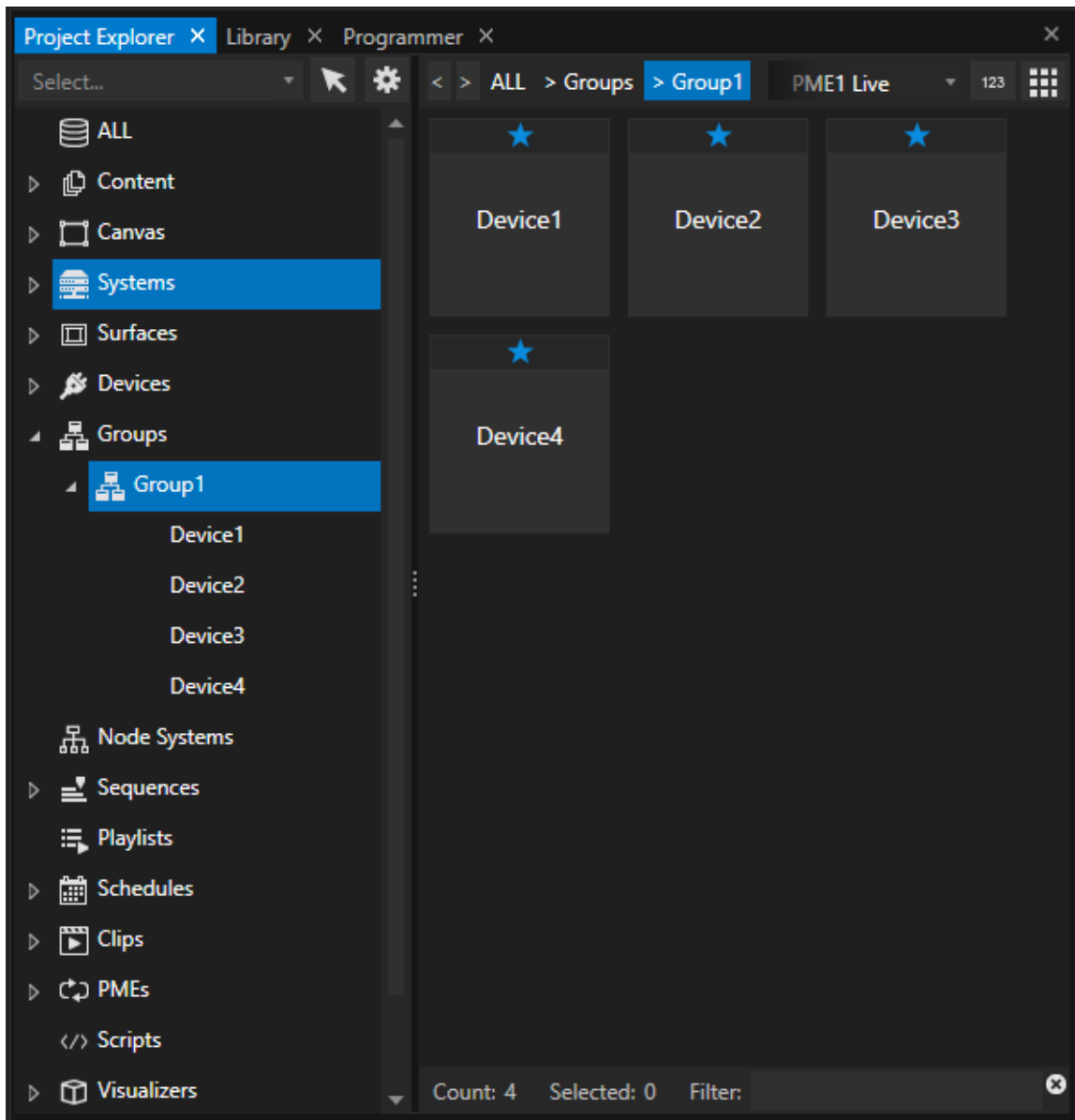
### 5.5.1.3 Groups

- Grouping is a good way to **effortless manage a bunch of devices**.
- **Control a large number of devices** with the help of groups.
- With **shared device groups** you are able to control all devices of a group from **only one clip container** in your sequence.



In the current assembly version of VERTEX the group functionality is supported for devices only.  
More will come with future releases.

## Create a group and assign Devices



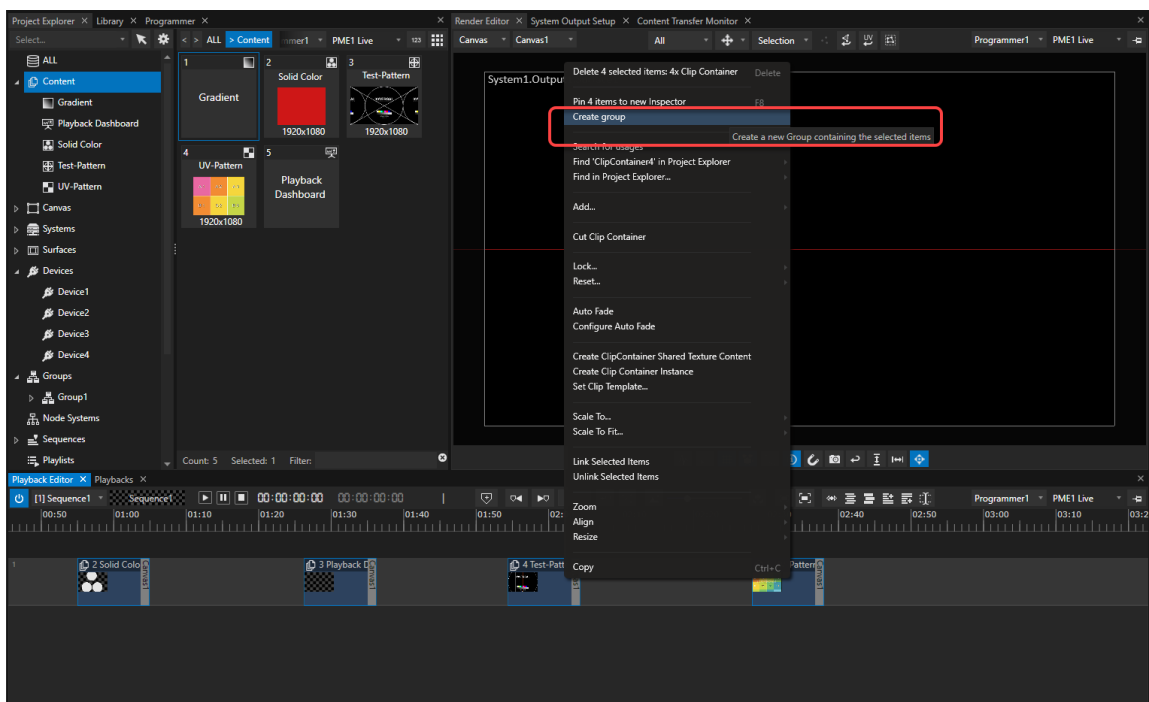
1. navigate to the group section in the project explorer
2. right-click with your mouse on **Groups** to open the context menu
3. select **Create Group**
4. a new group is created
5. rename this new group ( with help of the context menu or by pressing the shortcut "f2")
6. go to the device section of the project explorer
7. select all devices that should be into this group
8. drag them with your mouse to the group

or

1. go to the device section of the project explorer
2. select all devices that should be part of a new group
3. open the context menu with a right click
4. select **Create Group**
5. a new group is created - all selected devices are part of this new group

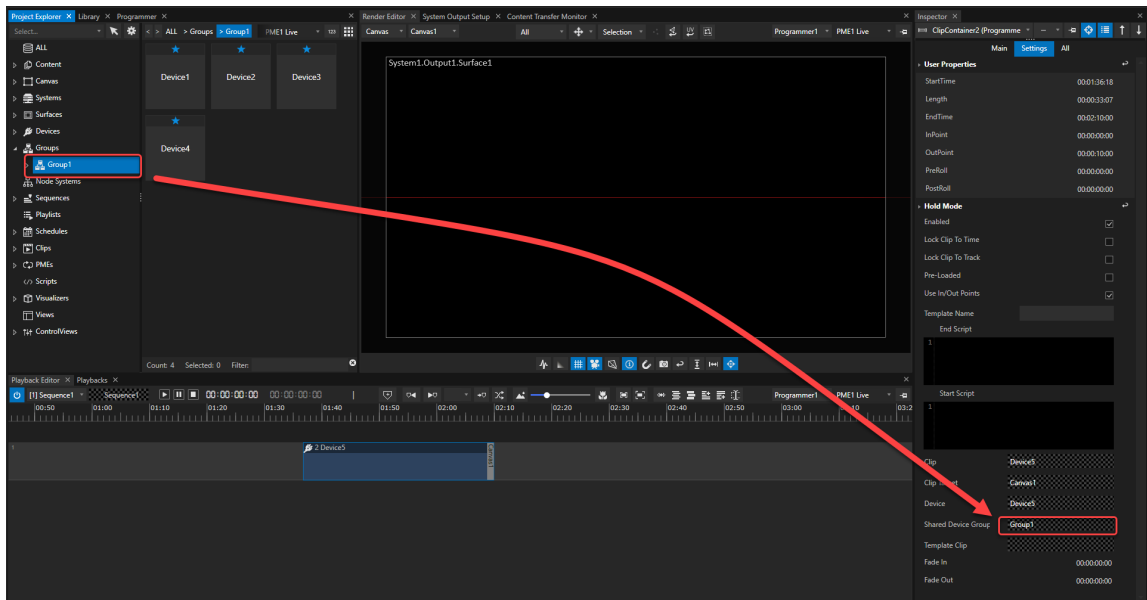
## Group of Clip Containers out of Playback Editor

- Select ClipContainers in the Playback Editor
- RightClick an open Context Menu
- Select "Create Group"
- A new group is created in the Project Explorer - Group Section
- All selected Clip Containers are member of this Group
- The Clip Containers are not linked to each other!



## Shared Device Group

- Shared Device Groups allow you to **control all items of a group from a single device**
- Shared Device Groups can be assigned for each **clip container of a device**
- Shared Device Groups **are assigned in the inspector for a device clip container**



## Set up a Shared Device Group

1. create a clip container for one of your devices: drag the device into the playback editor
2. select this clip container and
3. access its settings in the inspector
4. use your mouse to drag a group from the project explorer into the shared device group property field in the inspector
5. now all settings for this device clip container are shared with all other devices of the group
6. when adding new devices to your group, all settings are also shared with this new devices

### 5.5.1.4 Versioning

- VERTEX comes with a **unique version management** that allows you to **handle different content versions with ease**.
- **Import a new version into project explorer** and switch between the different versions of your content back and forth.
- **The version of every content file that is used** in your sequences and playbacks can be changed **with only one click in the inspector or by script**.

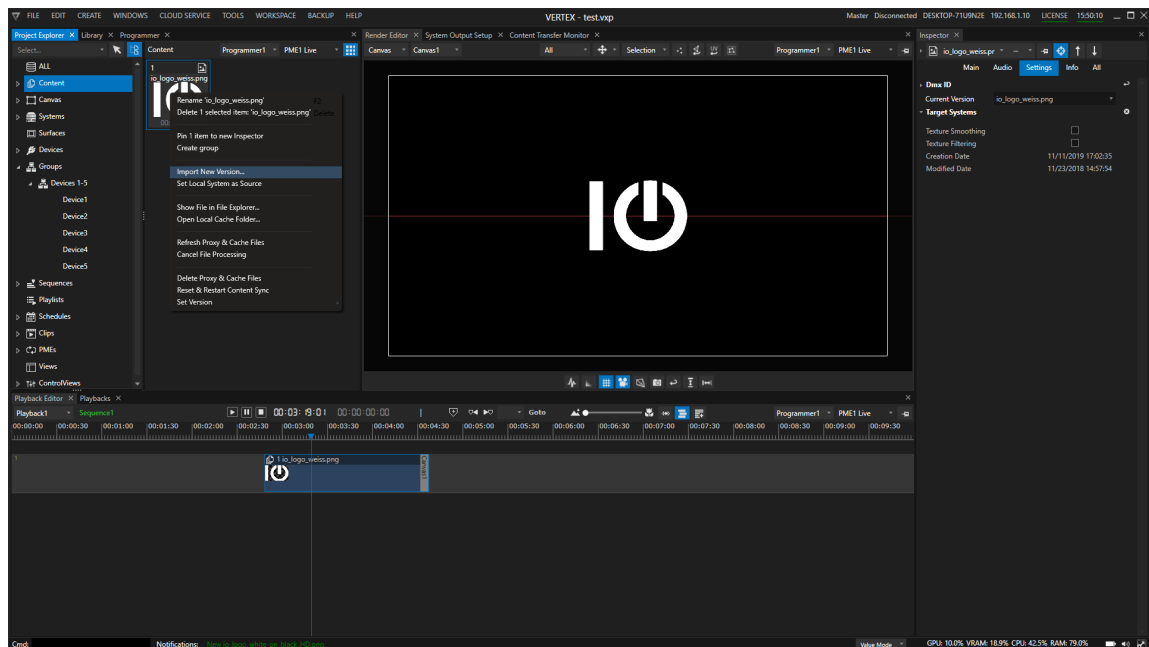


## Import a new Version

The **import of a new content version differs from the workflow** how regular content is imported into VERTEX

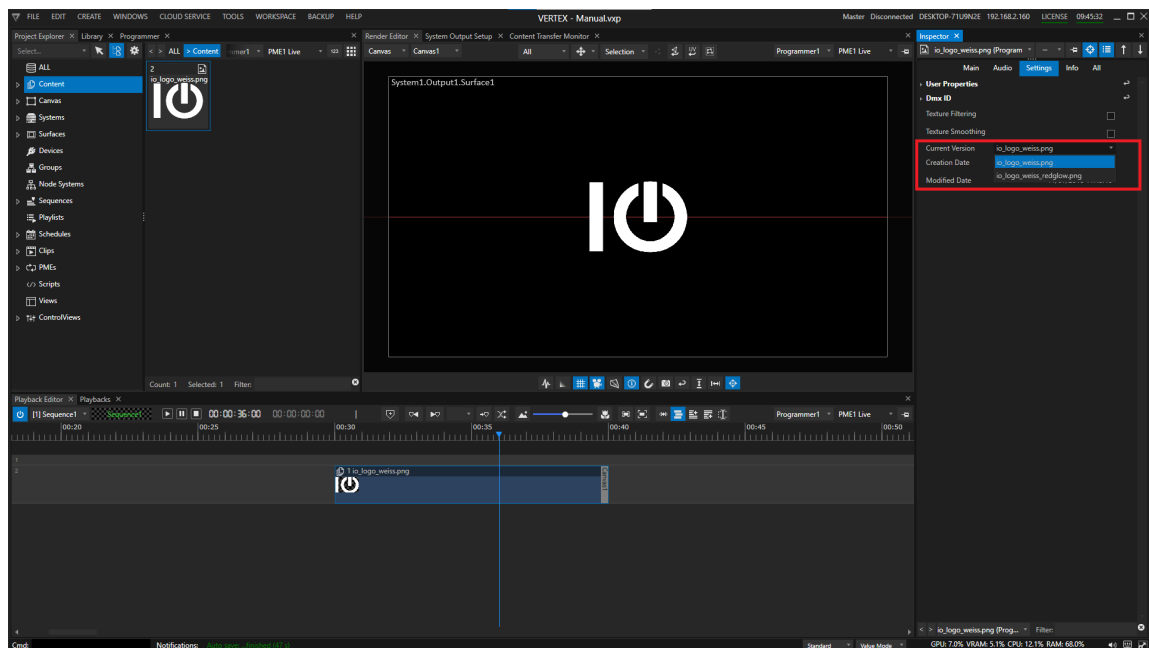
The command **Import New Version** is always related to an already existing content item in your project explorer.

Therefore you will find the **version import dialog only in the context menu** of the content item in regard.



1. select a content item in your project explorer
2. open the context menu by a right click with your mouse on this item
3. select "import new version"
4. a file explorer window pops up - select your new content there and confirm
5. proxy and audio files are generated for this content -
6. the thumbnail in the project explorer is switched to the new content
7. all clip containers or playback clips are changed to the new content.

## Switch back and forth between content versions



1. select the content item in the project explorer
2. the item should be shown in the inspector
3. go to the settings tab in the inspector
4. select your preferred versions in the content-version drop menu
5. the content has now been changed in all places of your project to the version you have selected.

Version switching also is possible with [Script Commands](#)

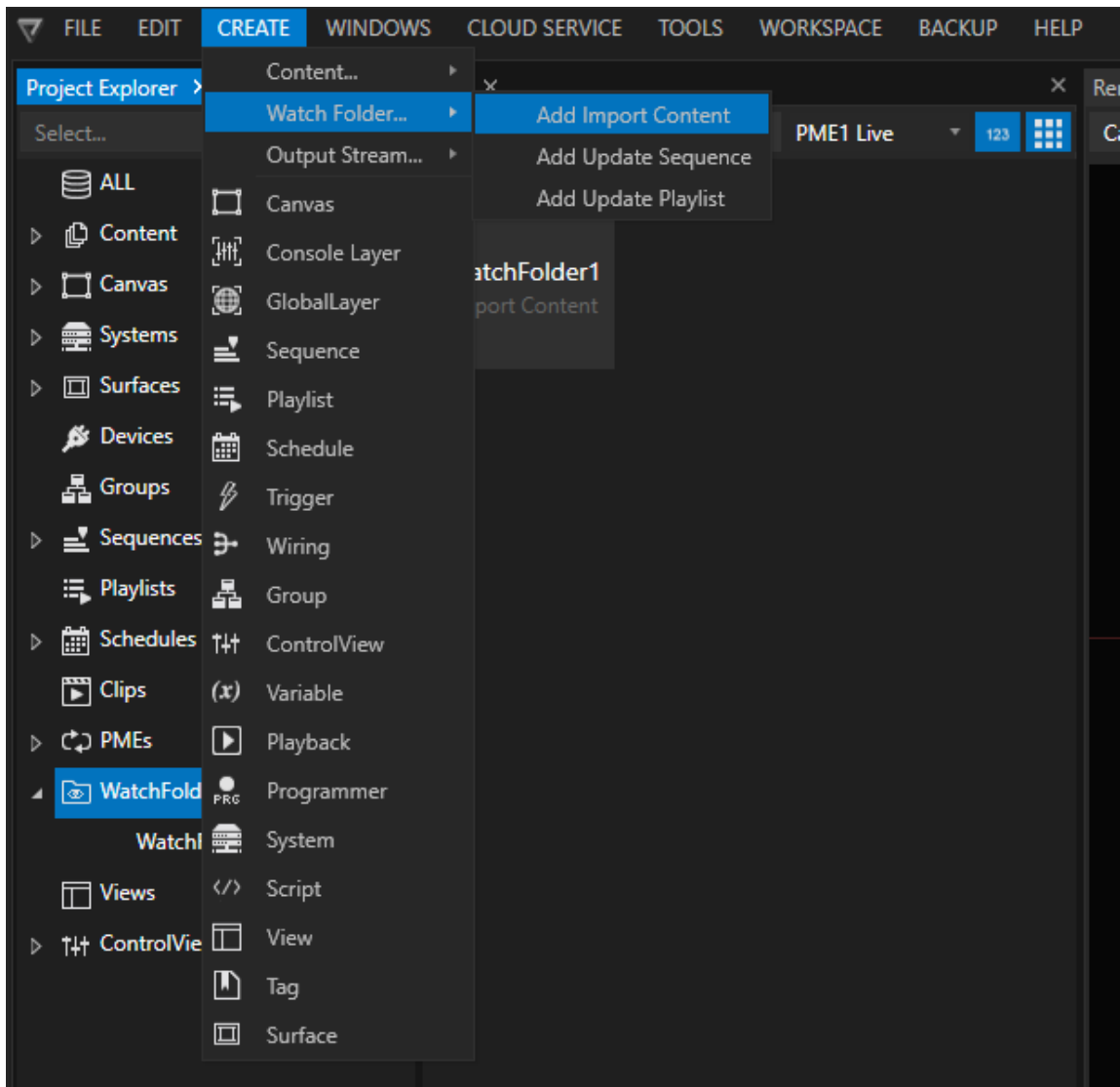
### 5.5.1.5 WatchFolder

- WatchFolders help you and **observe windows folders for file changes**
- **Content** from a predefined filepath is **automatically imported**
- The defined folder is **checked for changes on a given interval**
- **There are 3 types of WatchFolders: Import Content** (only imports content), **Update Playlist** (Import Content and automatically generates a Playlist) and **Update Sequence** (Import Content and automatically generates a Sequence)

## Create a WatchFolder

- Go to the **Main Menu**

- Open **Create**
- Select **"WatchFolder"**
- **Choose a type**
- Once a first WatchFolder is created, there is a **"WatchFolder"** Section in the Project Explorer

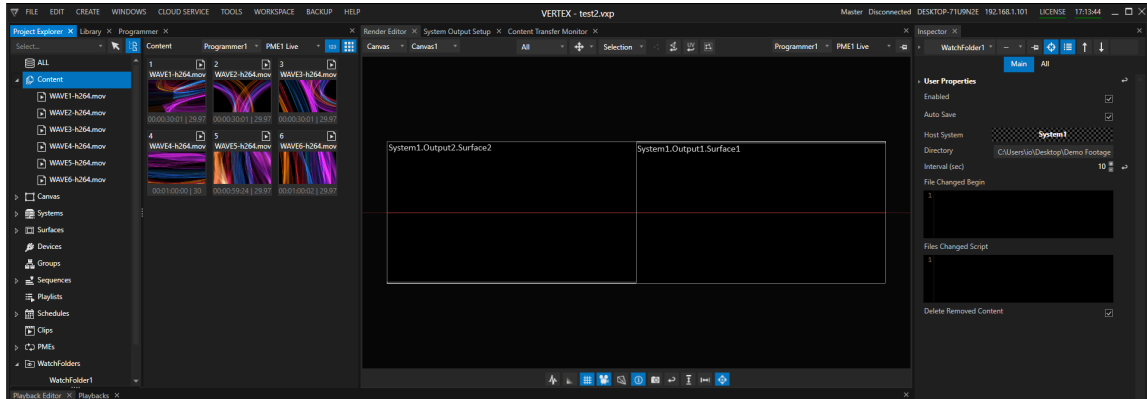


- Select a **WatchFolder** in the Project Explorer
- Do the **settings** into the Inspector

## Import Content

- The **"Import Content"** WatchFolder **automatically loads Content** into your Project.

- All content files into a Windows directory will be loaded as content items into VERTEX



## Settings

### Auto Save

*Saves the project when new content is imported*

### Delete Removed Content

*Removes content from the project if the files are deleted in the WatchFolder*

### Directory

*The local Windows filepath to the WatchFolder*

### Interval

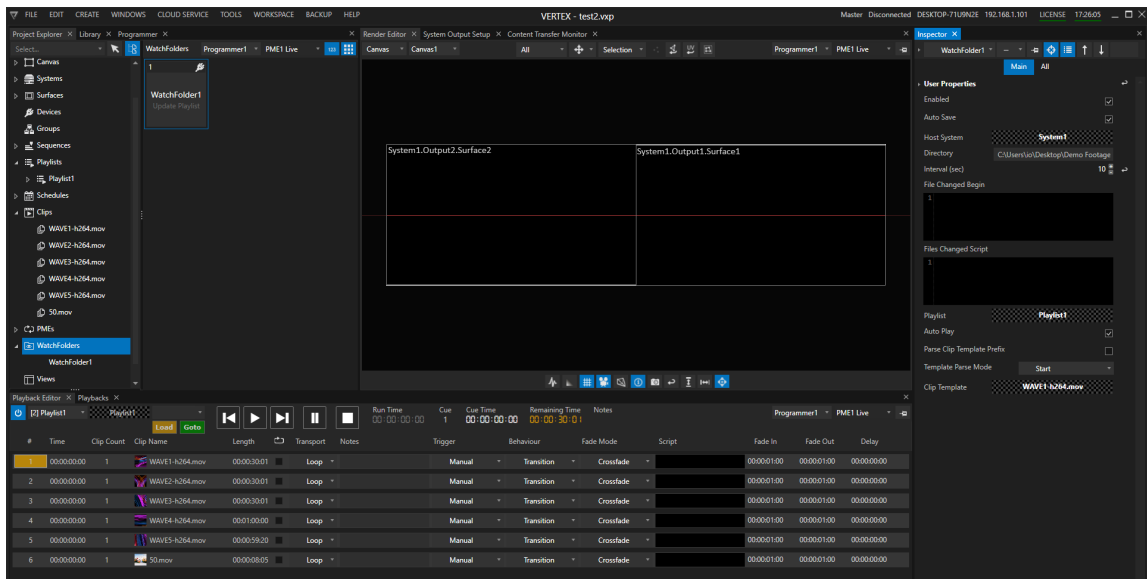
*Time in sec in which the WatchFolder is checked*

### File Change Scripts

*Scripts which can be executed either when a new file is detected (File Changed Begin) or when process of adding files is completed (File Changed Script)*

## Update Playlist

- The WatchFolder "Update Playlist" automatically loads content into your project
- All this Content items are added automatically to an empty Playlist.



## Settings

### Auto Play

Automatically sets the playlist to play

### Auto Save

Saves the project when new content is imported

### Clip Template

Select a [clip to act as template](#) for new imported contents

### Directory

The local Windows filepath that should be watched

### File Change Scripts

Scripts which can be executed either when a new file is detected (File Changed Begin) or when process of adding files is completed (File Changed Script)

### Interval

Time in sec in which the WatchFolder is checked for updates

### Parse Clip Template

Future-use, automatic template assignment by file name

**Playlist**

Drag n Drop an [empty playlist](#) to which clips from the WatchFolder should be added

**Template Parse Mode**

Future use

## Update Sequence

- The WatchFolder "Update Sequence" automatically loads content into your project
- All this Content items are added automatically to an empty Sequence

## Settings

**Auto Play**

Automatically sets the Sequence to play

**Auto Save**

Saves the project when new content is imported

**Clip Template**

Select a clip to act as template for new imported contents

**Directory**

The local windows filepath to the folder that should be watched

**File Change Scripts**

Scripts which can be executed either when a new file is detected (File Changed Begin) or when process of adding files is completed (File Changed Script)

**Interval**

Time in sec in which the WatchFolder is checked

**Overlap Time**

Crossfade duration for imported clips

**Parse Clip Template**

Future-use, automatic template assignment by filename

**Sequence**

Drag n Drop empty sequence to which clips from the WatchFolder will be added

**Starttime**

Set the timecode of the sequence at which the WatchFolder automation starts to import clips.

**Template Parse Mode**

Future use

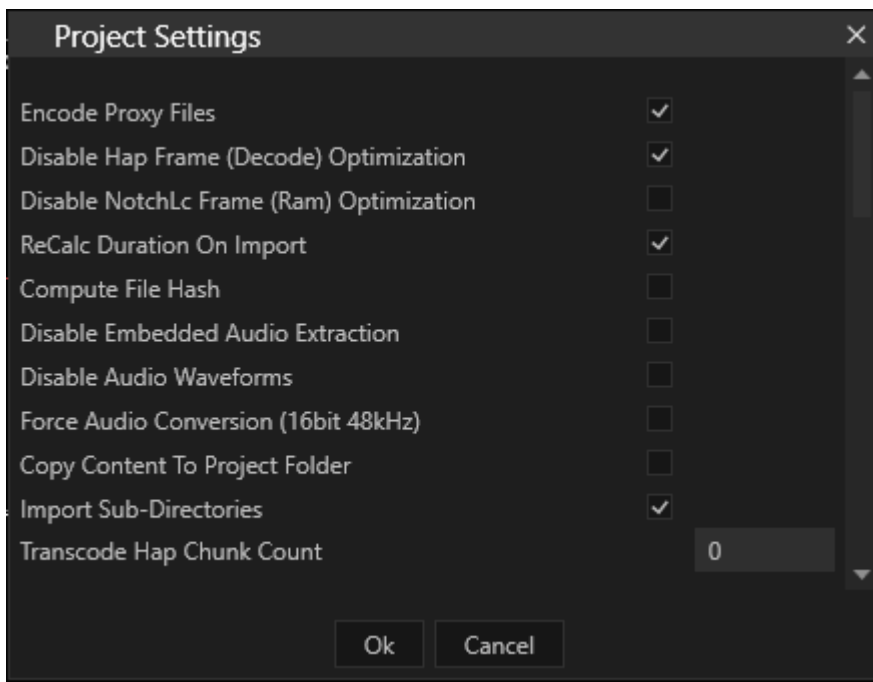
### 5.5.1.6 Target Systems

Settings for Target Systems only become relevant if you work with [several VERTEX systems in one project](#) and want to distribute some content to different systems.

The documentation for the advanced feature-set of VERTEX is in progress and will be updated step by step  
Until then: Please drop us an E-Mail with your "How-to-do-this-in-VERTEX" question to [support@ioversal.com](mailto:support@ioversal.com)

## 5.5.2 Project Settings

- Every VERTEX Project has the option **to set global parameters and settings that are valid for the whole project**
- **Be careful** - some changes could have massive effects on e.g. performance. ioversal ships VERTEX with some default project settings that are tested and set by experience
- **Some options are related to the workflow you prefer** - e.g. use always Top-Left Coordinates for a Canvas
- **Open the Project Settings window via the "Edit" Tab into Main Menu on Top.**

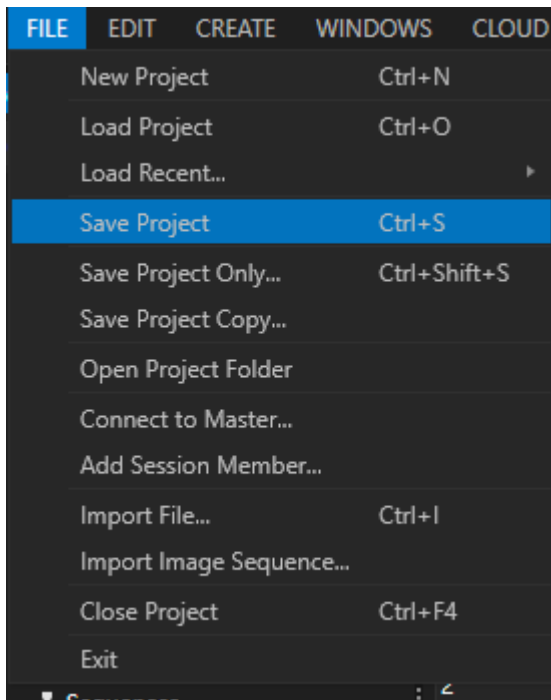


### 5.5.3 Load and Save Project

- Open **"File"-Tab into the Main Menu** to Load or Save a Project
- There are **different options to save a Project** - depending on your usecase and the data that is be included
- **Autosave** supports you during your work

## Load Project

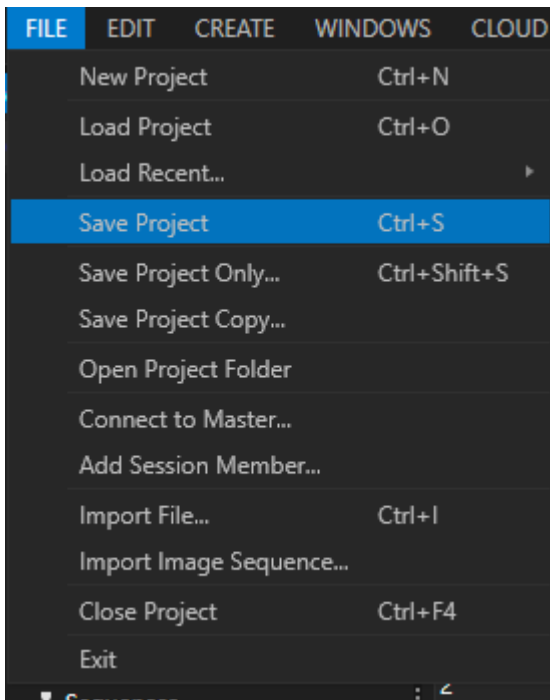




- **Load project** opens an Explorer Window to select a VERTEX project File that should be loaded. navigate to your Project Folder and select the \*.vxp File
- **Load recent** opens a list of the recent projects you have worked on.

Both options will of course close your current project. Before the project is closed, a modal dialog ask you if you want to save your current work.

## Save Project



1	<b>Save Project</b>	<i>Saves Project at regular location of the Project File. Use Shortcut CTRL + S to save.</i>
2	<b>Save Project only</b>	<i>Saves Project File only. Opens an Explorer Dialog to select file location and name. Does not Save the Data Folder of a Project. Use e.g. to Save different project Versions into the same project folder, as a personal backup or if you need only the project file for e.g. support requests</i>
3	<b>Save Project Copy</b>	<i>Saves the whole Project Folder - including Proxy Files, Content and Data. Opens an Explorer Dialog to choose a destination and folder name. Take care: depending on settings and assets of your project a bigger amount of data could be saved. Good for personal backups or to exchange a project copy.</i>



### **Save Project on a Session Member**

When working on a Session Member, the save options are different. You have a choice between "Save on Master" and "Save on Local System". Recommended option is to use "Save on Master" and trigger a project save on the Master System.

Session Members always pull the project data from a project file that is opened from the Master System.

Use "Save on local System" to create a local project file or backup on the session Member

## Auto Save

- Autosave supports you during your work and **automatically saves your project**
- By default, **autosave is triggered every 5 minutes**. This interval **can be changed** in the [Project Settings](#)
- Autosave is **saving up to 4 versions of your project**. After this number is reached, the oldest version is deleted
- The files are **located** in your Project Folder in **subfolder with the name "\_bak"**
- The autosave-files all **have a file suffix "\*.vas"** - (VERTEX Auto Save)
- To **recover your project**, please rename the file suffix to .vxp or just open the file with VERTEX.

## 5.6 Multi System Configuration

### Prerequisites for a VERTEX Session

1. VERTEX has to run on all Systems in the **same Assembly Version** such as "VERTEX 2023 R1 2023.01.14.0439"
2. **All Systems must have a valid license.**
3. **Network Adapter Settings** and **Default Datasync Adapter** must be set for all systems to the same network.
4. **Vertex Background Services** must run on every System.

### VERTEX in Session Mode

A VERTEX Session always consist of

- **ONE Master System** where the project file is created, loaded and saved.  
and

- **ANY NUMBER OF Session Members.**

Session Members are VERTEX systems that connect to the Master System's project.

The number of Session Members is designed to be unlimited, flexible and scalable:

it is possible to add new Session Members to a project or delete existing Session Members.

### The 5 Cogs in Session Mode:

#### 1. **Data Sync**

All project data and settings are shared between all systems in real-time, thus creating a true multi-user environment.

Every system will have the same data status, regardless on which system a change has been made to the project.

#### 2. **Playback Sync**

Playbacks are always in sync on all systems in a VERTEX Session.

Any system (both Master or Session Member) can be defined as the clock master and will be synchronizing all playbacks of every Session Member.

### 3. [Content Sync](#)

VERTEX Content Sync is designed with a peer to peer technology that results in faster data transfer with increasing number of systems added.

Content can be imported on any system in the session and even so, all data is being shared in the background between all systems.

### 4. [Data Routing](#)

Incoming continuous data is shared between systems in a Session and can be routed: For instance, Incoming DMX-Data on System1 could be routed

as Art-Net™ output on System3. Incoming device data on one system is also shared between all other systems.

### 5. [Backup Scenarios](#)

This modular method of connecting Session Members is the foundation for VERTEX's [flexible backup possibilities](#): create a redundant backup or only a partial one.

Session Members can switch render roles and render for another system that drops out. The Master role can also be taken over by another Session Member.

[Information on Session Mode's advanced settings can be found here.](#)

## 5.6.1 Session Management

- Set up a Session with **multiple VERTEX Systems** into a Project: a **Master** and a number of **Session Members**
- The **Master manages the Project Data**. Settings between Session Members and Masters are shared. The **Project File is stored and loaded on Master**
- You are able to **connect and to disconnect a System**.
- **Systems** could be **virtual** or **real**

## Master and Session Member

When working with **multiple Systems** into a VERTEX project, each VERTEX license can take **a role called "Master" or a role called "Session Member"**

This role does not depend on the license and can be selected for each new project.

The **Master manages the Project File and all changes**. The system on which a new project was created

gets the Master Role by default.

**All other Systems** that will be connected and added to the Project **are Session Members**.

**All Data is shared between Session Members and the Master. A Master is responsible for handling of this Data Exchange.**

The Master is not necessarily responsible for the [Sync Clock for Playbacks](#).

**Session Members** can work in **Fullscreen** mode, in **only UI** mode or in **a combination of both**.

You can work with different users together in one Project. All Session Members are allowed to make changes an every Property.

## Data and Project Handling

The **Project File** is loaded and stored on Master. After a Project File is loaded on a Master, the **Master System pushes all Project Data to the Session Members. Content that is already distributed to Session Members is not shared again.**

The **main Project File** is always saved on the Master.

**Session Members** are able to trigger a Project **"Save on Master"** or to save a **"Local Copy"** of the Project File.

It is **not possible yet to merge offline changes from Session Members to a Master Project.**

Please work on the main Project File from Master and Copy this Project File back to Master again.

When open a VERTEX Project with a System that is not a Master, VERTEX will ask you if you want to map your local System as the Master.

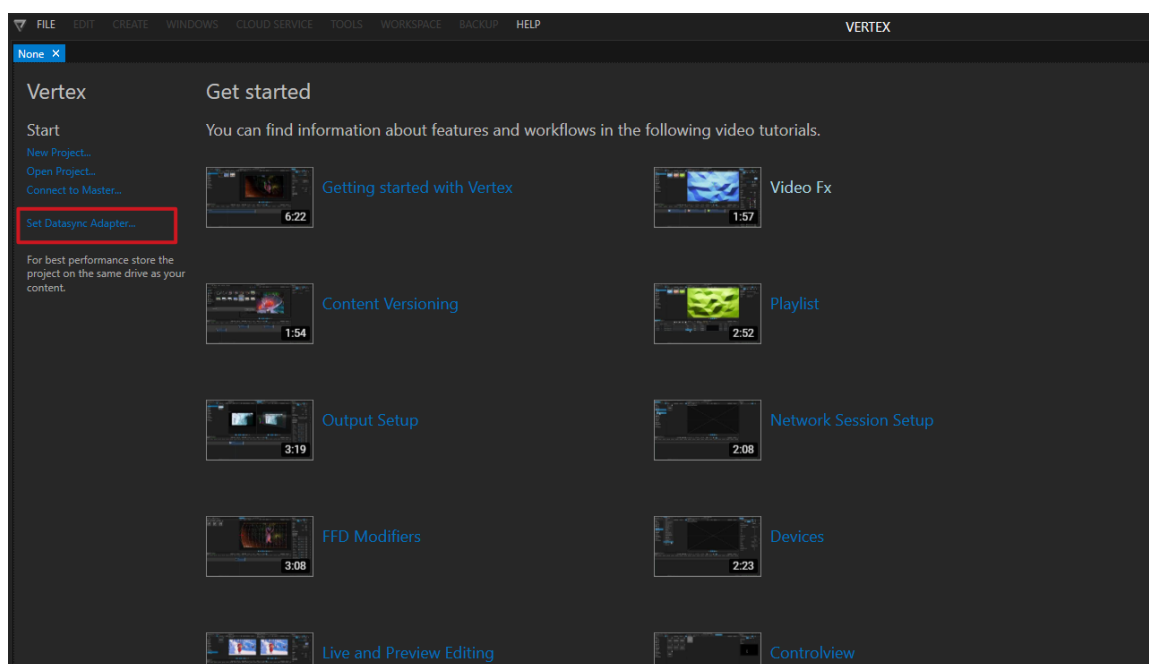
## Prerequisites for a VERTEX Session

1. VERTEX has to run on all Systems with the **same Assembly Version** - e.g. "2020\_Q1\_2020.05.20.1920"
2. **All Systems** must have a **valid license or no system may have a license** and all run with the Trial
3. **Network Adapter Settings** are done - The **Default Datasync Adapter** is set for all Systems into the same network.
4. **Vertex Background Services** must run on every System

## Network Adapter Settings

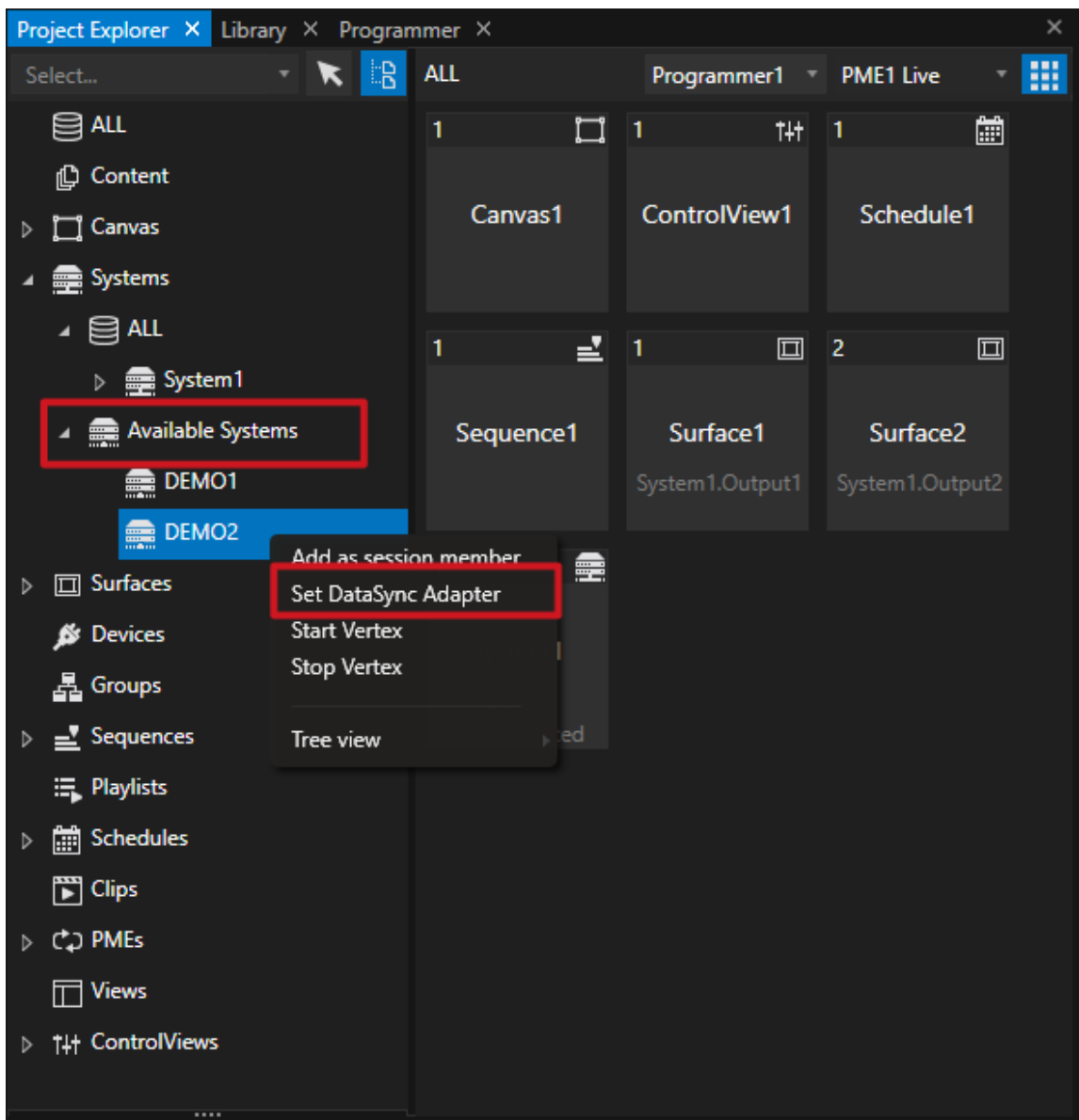
For each of the Systems, the **Default Datasync Adapter** has to set once on every System

You can do this for a Session Member **with a link on the Startup Page**

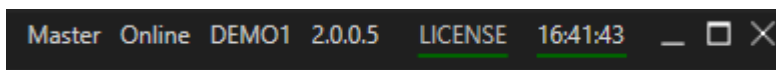


Or on a Master in the "Available Systems" Section of the **Project Explorer**.

Right-click on a detected VERTEX System to open the Context Menu and set the Adapter.



Once the adapter is correctly set and recognized by VERTEX, the **IP address** will appear in the Top Bar





**Reboot after Adapter or IP change**

We recommend to first reboot your Windows System after an Network Adapter was changed into Windows 10.

In some cases the VERTEX Background Service can only detect the changes after the System is rebooted

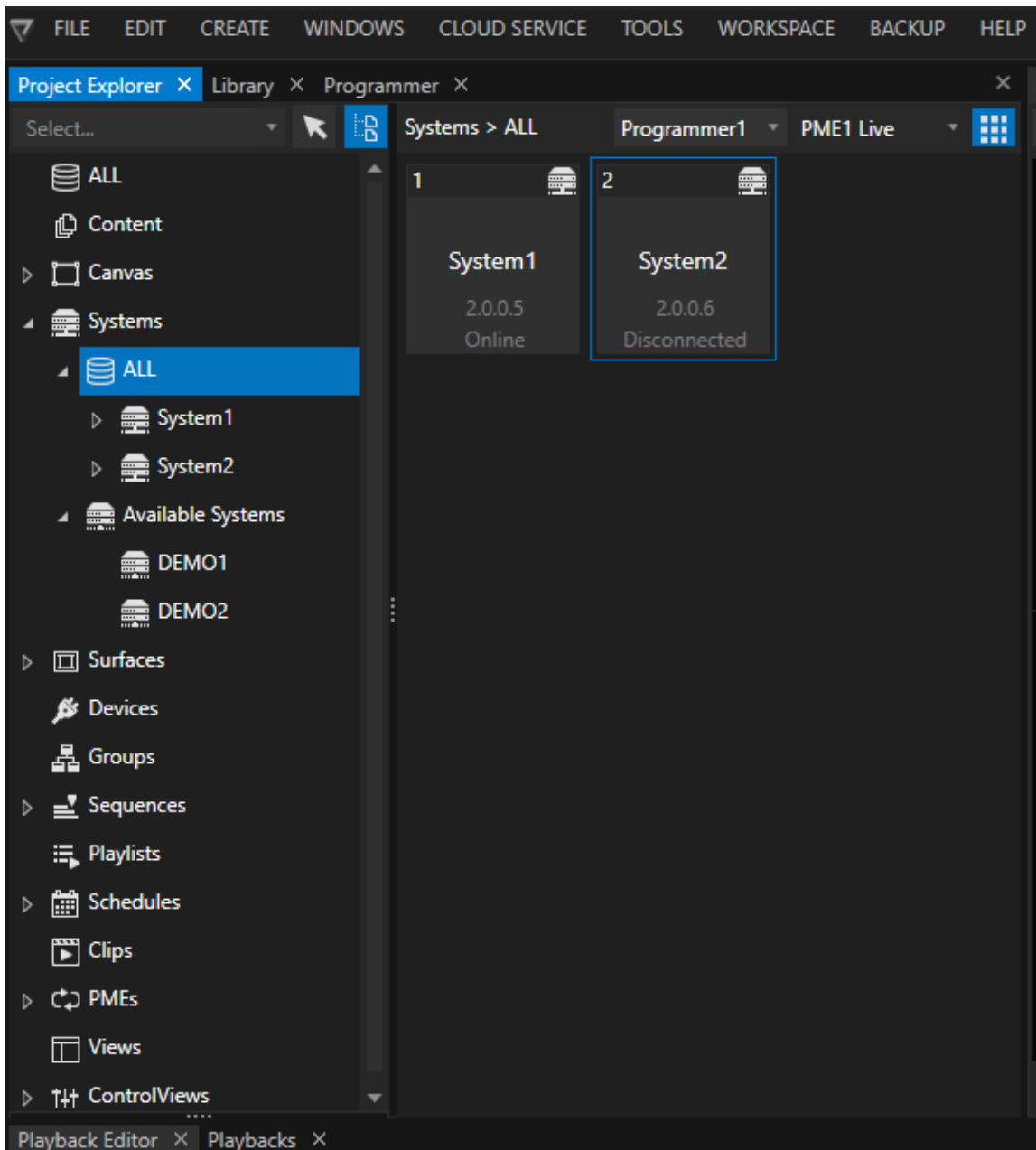
## Add a System as Session Member

### Add a System from Master

Go to **Systems Section** of the **Project Explorer**. **Drag** with your mouse **a System** from the **Available Systems** Section to **"All"**

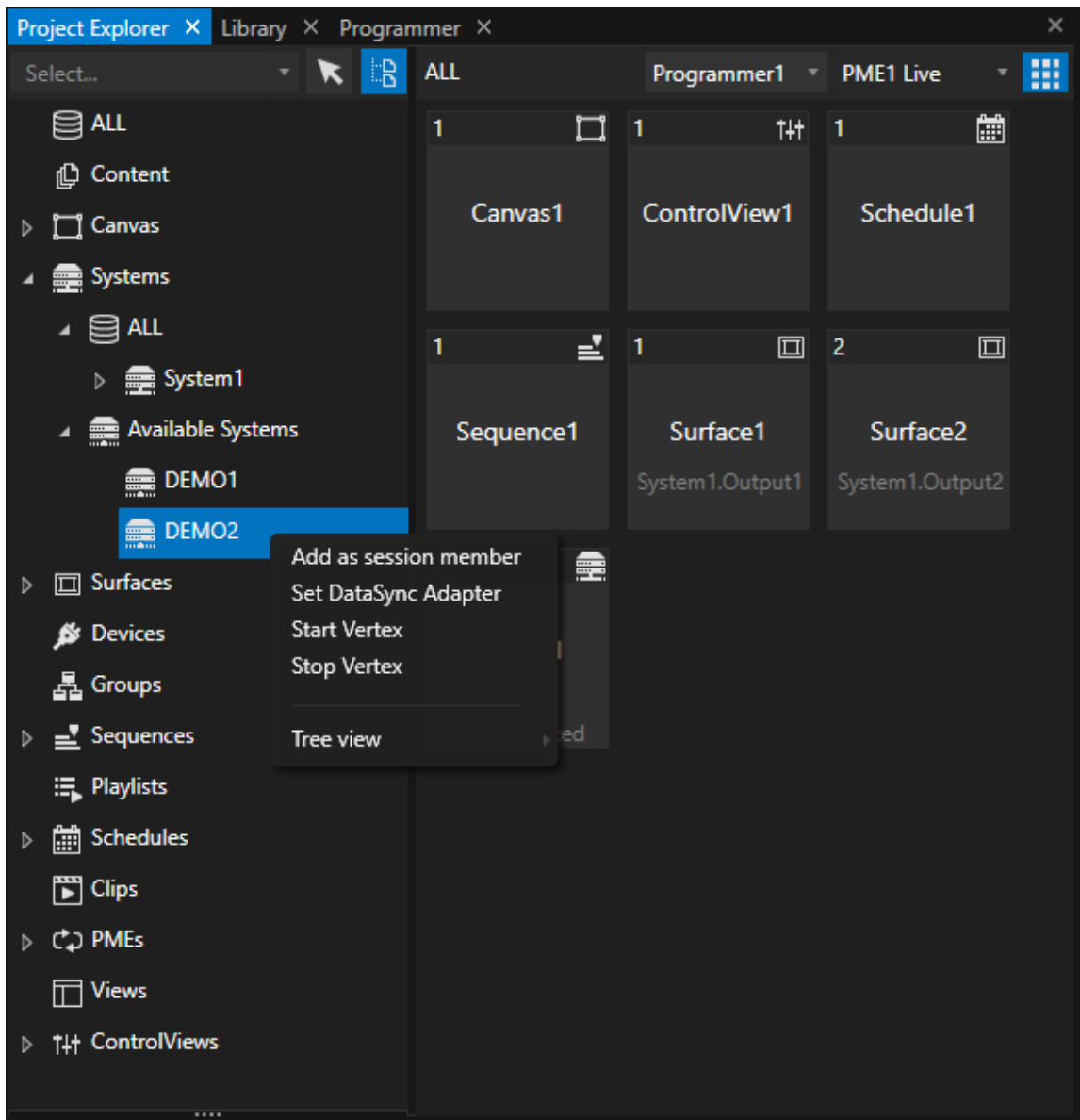
If the [prerequisites](#) are all given, the new system appears in the "all" section and connects automatically.

If not, there should be a notification in the Status-Bar



You can also use the **context menu**.

Just **right-click on a System** listed under **Available Systems** and choose **"Add as Session member"**.



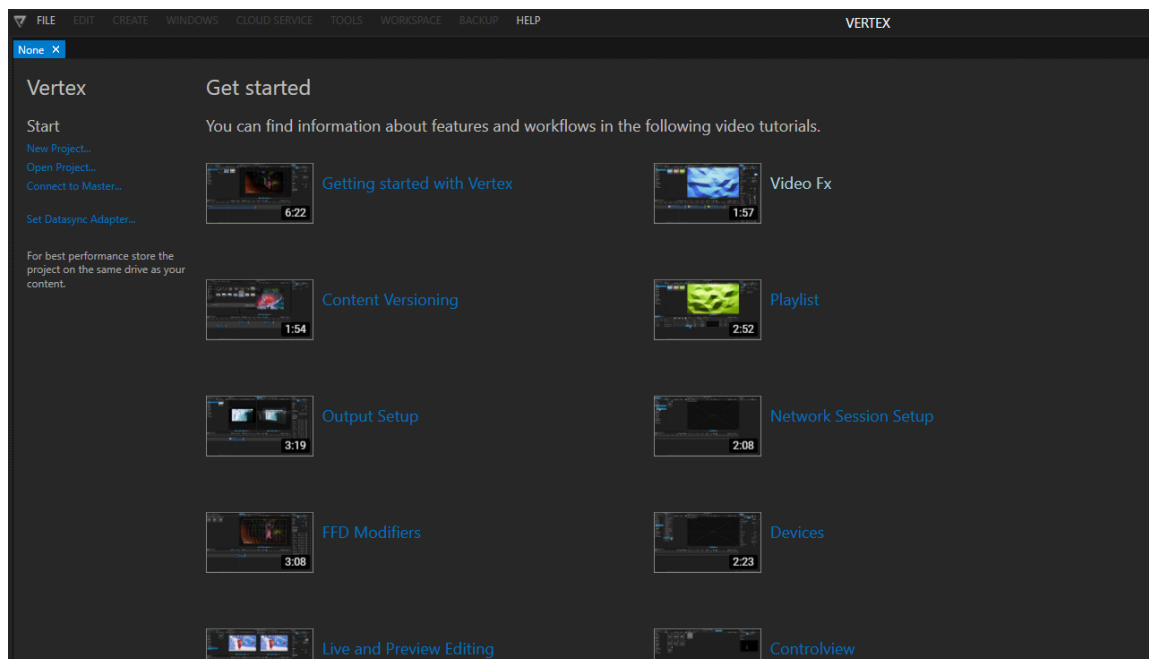
**Connect with IP Address if System is under "Available Systems"**

If for some reason a System is not listed under Available Systems, you can connect it with its IP Address

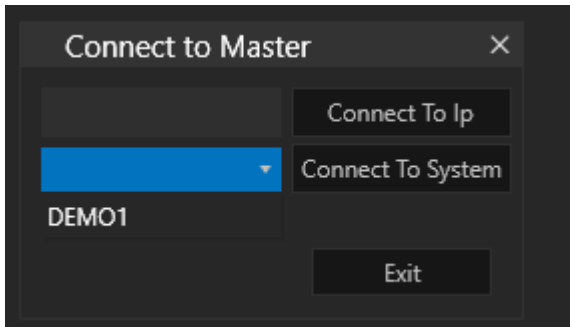
Go to File -Tab into Main Menu and choose "Add session Member" A dialog Window opens where you can enter the IP of the System

## Join a Project from Session Member

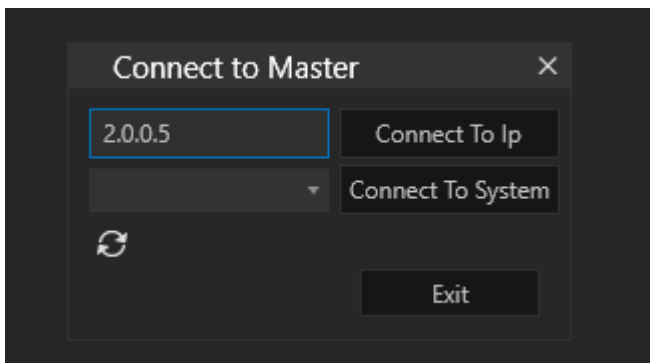
Start Vertex on a Session Member. Click to **"Connect to Master"** on the Startup Page



A window opens. Select an **available Master** from the **Dropdown-List** and confirm with **"Connect to System"**

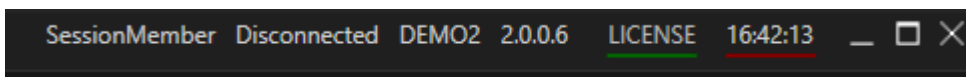
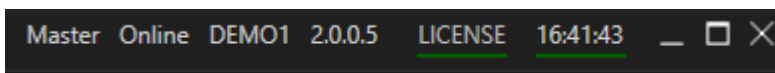


If no Master is displayed into the Dropdown, you can **connect to the IP**. Just enter the **Masters IP Address** and confirm with "Connect to IP"



## Role and Connection Status

Once a system was successfully added to a Project, its **role** (Master or Session Member) and the **Connection Status** (Online or Disconnected) is shown in the **Top-Bar**



## Connect, Disconnect and Autoconnect

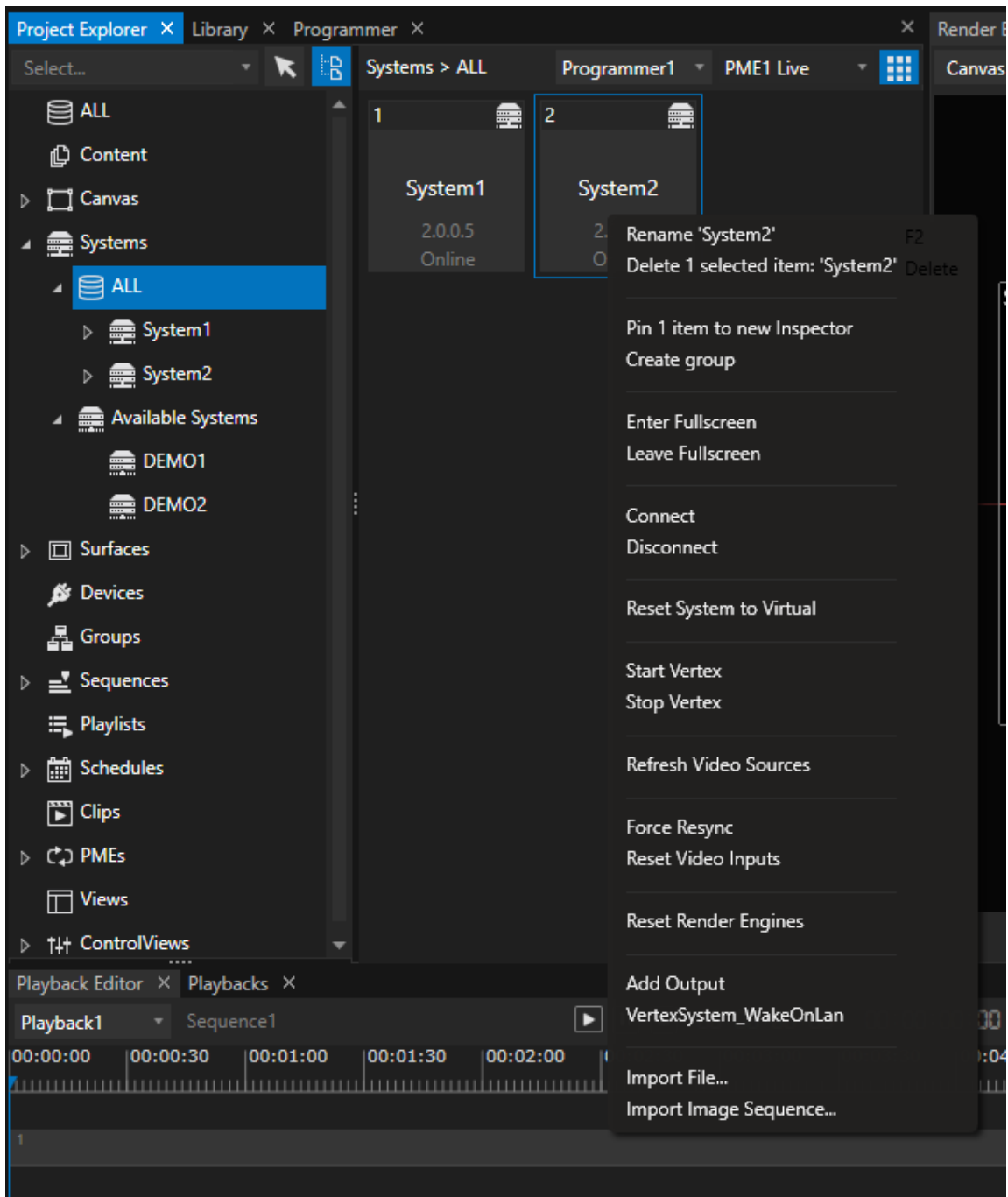
*By default **for all Session Members "Autoconnect" is enabled.***

*As soon as a Project is loaded on a Master and the VERTEX is running on a Session Member System, The Session Member connects automatically and loads the Project.*

*Autoconnect **can be disabled for each System in the Inspector Settings***

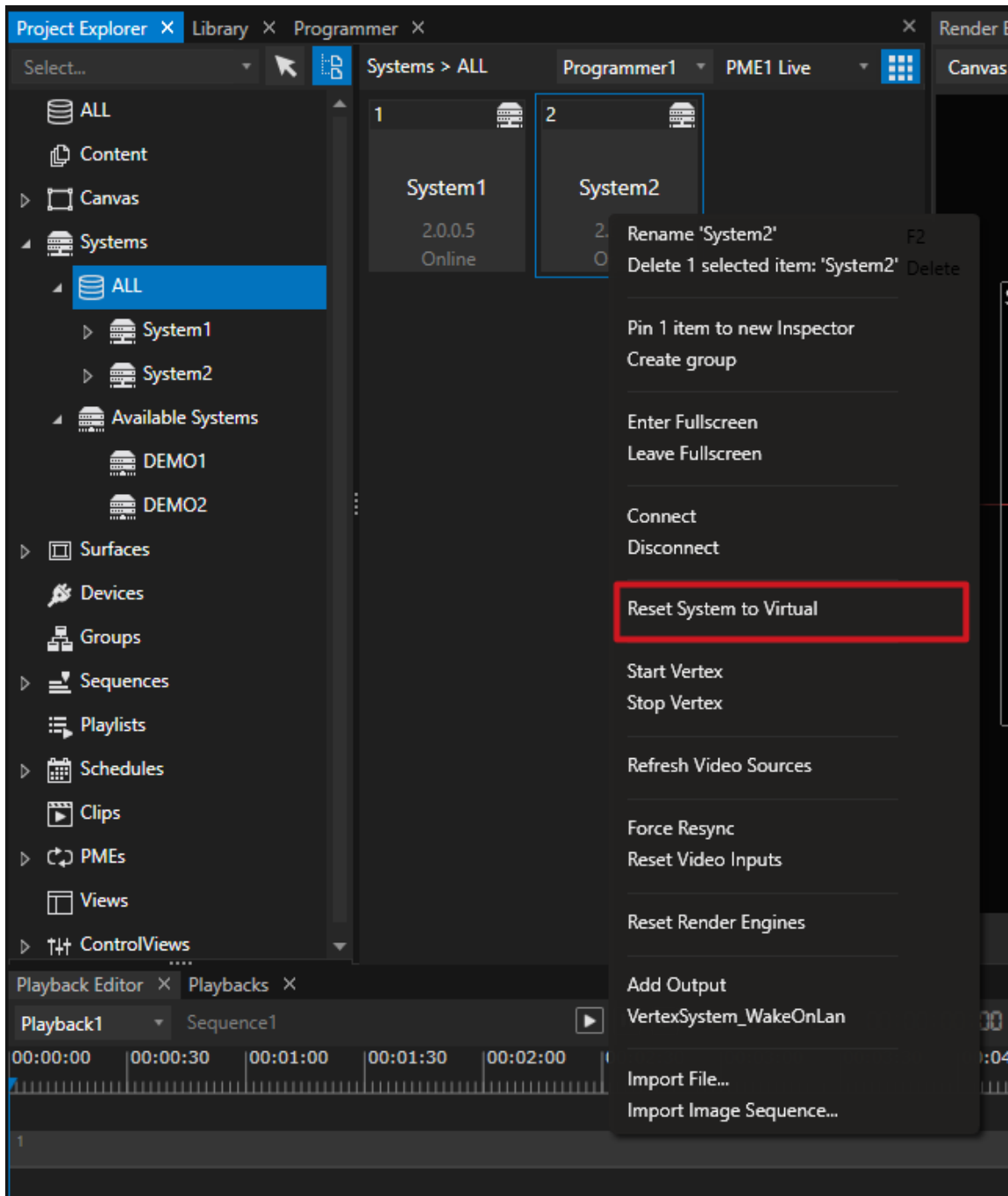
*You can **connect** or **disconnect** Systems **manually.***

***Right-Click on a System** and select "Connect" or "Disconnect" from the Context Menu*



## Virtual System

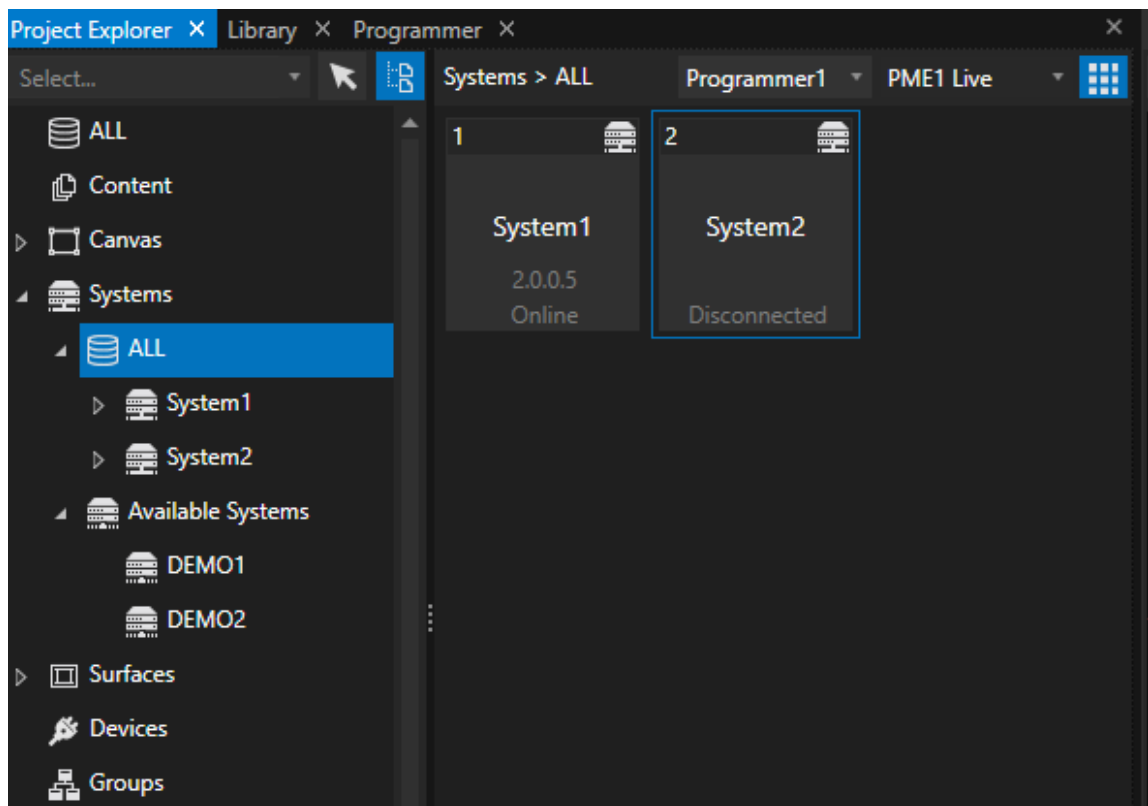
**Systems** in VERTEX also **can be virtual**. Virtual Systems are not linked to a real System into Network. You are able to switch a System into your Project back to virtual. **Unlink it from a System** by using **"Reset System to Virtual"**



**Virtual Systems** always have the **connection status Disconnected**

To link a Virtual System to a real one, just drag a System from the "Available Systems" list to to Virtual System

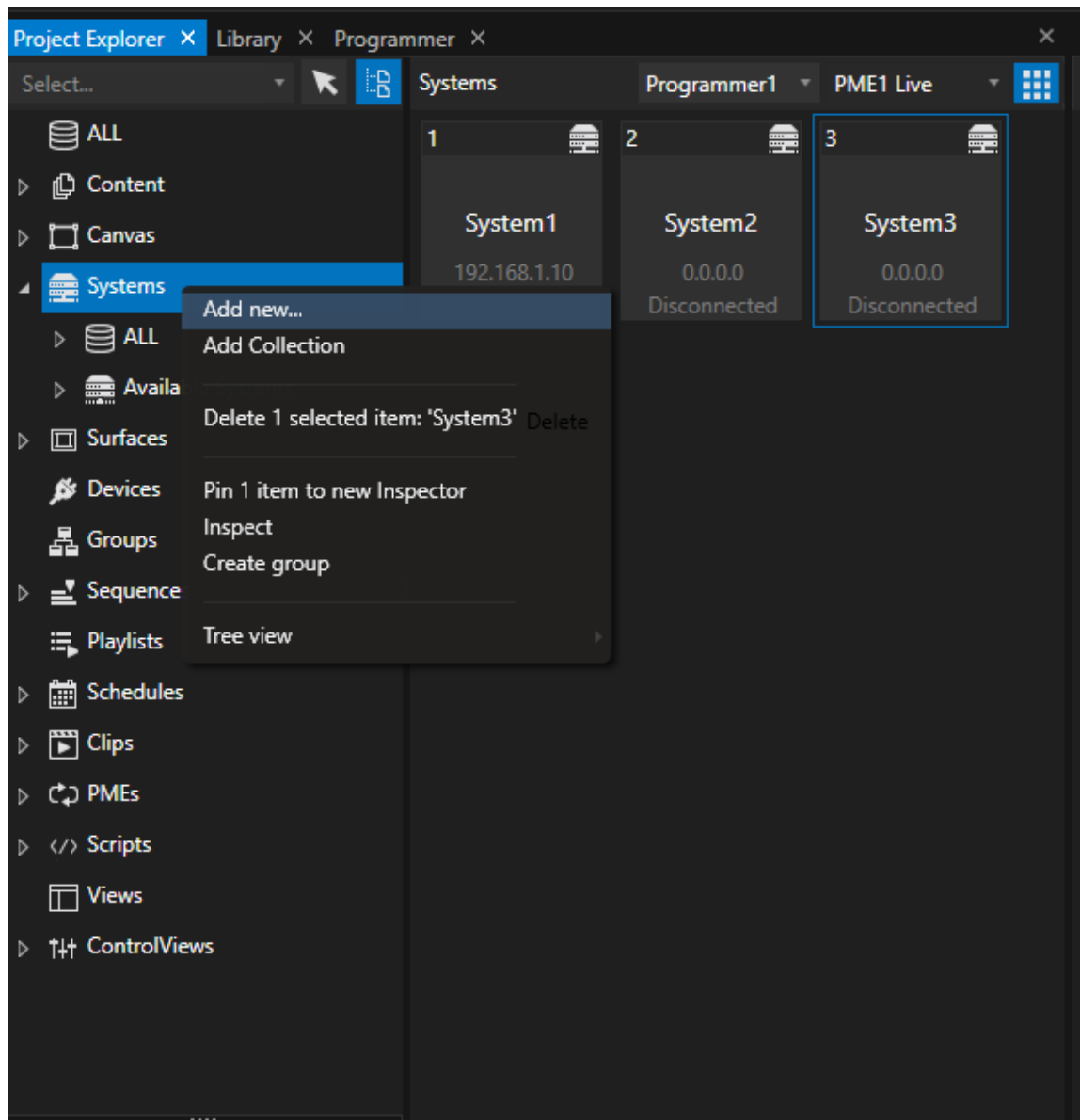




You may also create **new Virtual Systems** for your Project.

Use them for preprogramming or preview, later **convert them into a "real" System**.

Open **Context Menu** on the **Systems** Section of the **Project Explorer** and select **"Add new..."**



## 5.6.2 Data Sync

- The VERTEX Data Sync ensures that **all data is shared over all Systems into a Project**
- All Settings into all Editors are shared across all Systems. **Every System has the same data status.**
- This mechanism makes VERTEX to a real **Multi-User Environment**: Changes on Session Members and Master could be done in parallel by different users.

## Network Adapter Settings

- Please ensure, that for all Systems into your Project, the **Default DataSync-Adapter** is set.
- Read more about Adapter-Settings [here](#)
- We recommend a fast Ethernet-Network for VERTEX Data-sync

## Data Sharing over all systems

The unique VERTEX Data-Sync makes sure **that all data is available on all systems**. if you change a setting on the master, this setting is directly available on the session member. The same for Playhead Position when you do a Warping.

Data Sync works in the background and is **deeply embedded in the system architecture**. you don't have to worry about anything

## Multi User-Environment

The VERTEX Data-Sync **is the backbone to ensure a true multi-user experience**.

Data is shared over all systems - that also means that User 1 on System A directly can see changes that are made by user 2 on System B.

### 5.6.3 Playback Sync

- The VERTEX **Playback Sync** ensures that **all Systems into a Project are clocked and play out Content in sync**.
- The **Sync Clock Source System** generates the **Sync Clock** for all other Systems into a Project and **can be freely defined**.
- For **backup scenarios** there is an option to define a **Backup Sync Clock Source System**

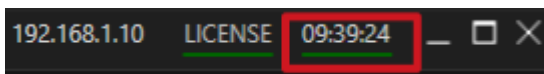
## Sync Clock

The Playback-Sync generates a **Sync Clock** that is **shared over Network to all other Systems** that are part of the Project.

This Sync Clock **ensures that all Playbacks over all Systems are accurately in sync**

The Sync Clock controls all Timecode and the Playheads positions.

When working in Session Mode a **status bar** shows you **if the system receives the playback sync** and is connected to it.



**Green:** Playback-Sync is received - System is synced

**Red:** System is not receiving the Sync Clock from Playback-Sync

## Sync-Clock Source System

For a simultaneous Playback into a mesh of VERTEX Systems **one of this Systems has to be responsible for generating a Sync Clock-**

This System is called the **Sync Clock Source System** and generates a **System Clock** which clocks all other Systems into your Project.

**By default the Master** is set as Sync Clock Source System. You are allowed to **change the Sync Clock Source System to any other System of your Project.**



### Sync clock and Master role

VERTEX offers great flexibility due to the mesh network structure. this is also reflected in the sync clock: the main clock is not bound to the master. The master can be the system that specifies the clock, but it does not have to be. The clock generator is not bound to the master and can also be taken over by any other session member. In this case, the Session Member is the Sync Clock Master.

## Example

A Project with 1 Master and 2 Session Members,

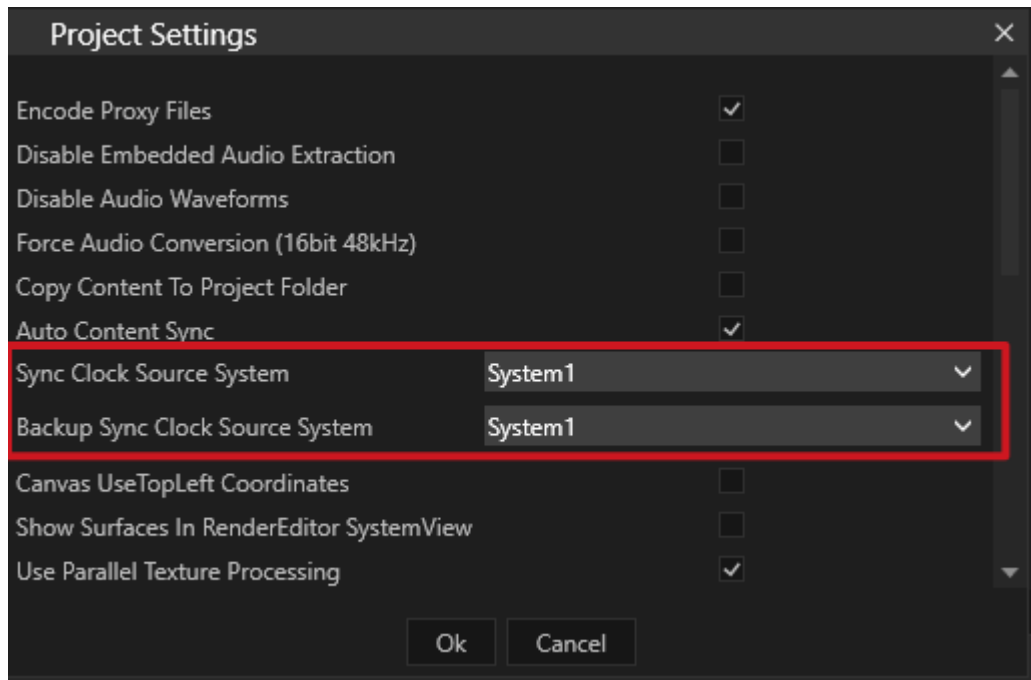
The ASIO Interface for [Audio Payout](#) is connected to System 2 (a Session Member)

The Sync-Clock for all Playbacks should be locked to the ASIO Clock.

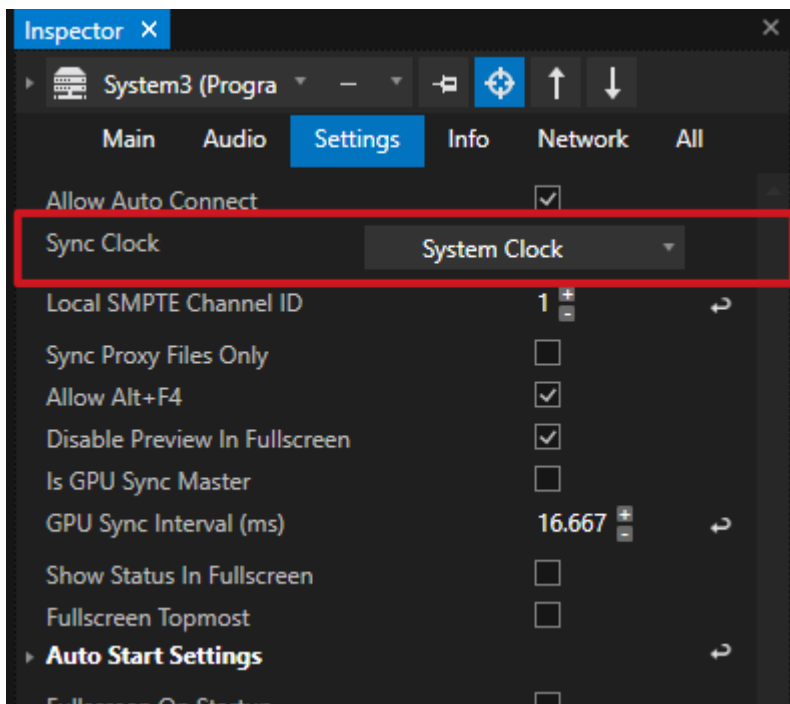
In this Case System 2 should be the Sync-Clock-Source-System for your Project.

## Set Sync Clock Source System

The Sync Clock Source System is globally defined in the **Projects Settings**.



## Sync Clock Settings



For every System into your project there is a **Sync Clock Setting**.

**By default** this setting is set to **System Clock**.

If there are problems with your Sync clock and simultaneous playback, please check the Sync Clock settings for all Systems

## Example

Let's continue with the Example from the previous Topic:

A Project with 1 Master and 2 Session Members,

The ASIO Interface for [Audio Payout](#) is connected to System 2 (a Session Member)

The Sync-Clock for all Playbacks should be locked to the ASIO Clock.

In this Case System 2 should be the Sync-Clock-Source-System for your Project.

**The Sync Clock Settings in this case are:**

**Master:** System Clock - should get the System Clock from Sync Clock Source System

**System 2:** ASIO Audio- System 2 is the Sync Clock Source System and should generate the System Clock from ASIO Clock

**System 3:** System Clock: should get the System Clock from Sync Clock Source System

## 5.6.4 Content Sync

- The VERTEX Content Sync is based on a **Peer-to-Peer File sharing technology** and it **becomes faster the more systems are connected**.
- Content is distributed **seamlessly and automatically to all Systems of a Project** - The **Content Transfer Monitor** gives information about the progress.
- The Content Sync works **bidirectional**: You are allowed to **import Content on any System** of your Project - not only on Master.
- **Restrictions** and certain **Target Systems** could be set **individually**

## Peer to Peer Sharing

The VERTEX content sync is based on a Peer-to-Peer sharing technology. Data is cut into little chunks that are exchanged between Systems

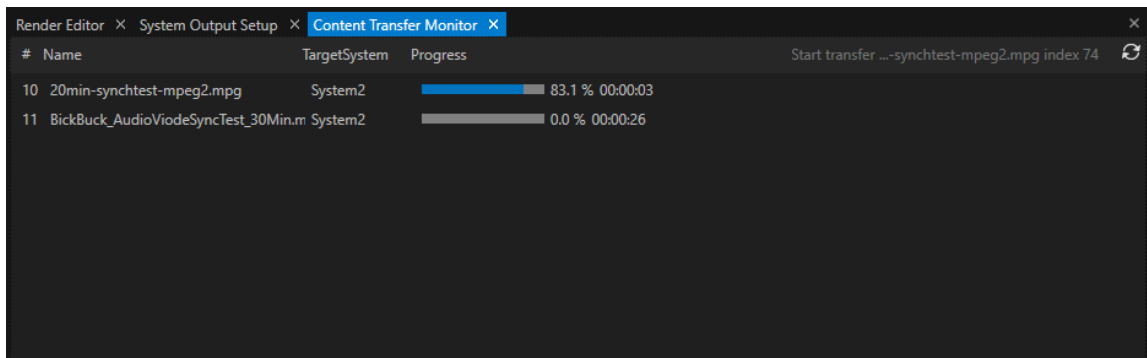
The distribution becomes faster the more Systems are connected and the larger the mesh network becomes.

All Data is stored into the VERTEX Project Folder that could be set for each System. If not set, the default Project Path will be taken into account.

## Workflow

1. **Add System(s) as Session Members** to your Project
2. Check **Local Project Path** for each System in the Inspector
3. Do settings in the Inspector for each System **that only should receive Proxy Files**
4. **Import your Content** on a System
5. Let the VERTEX **Content Sync work in background** and lean back...

## Content Transfer Monitor

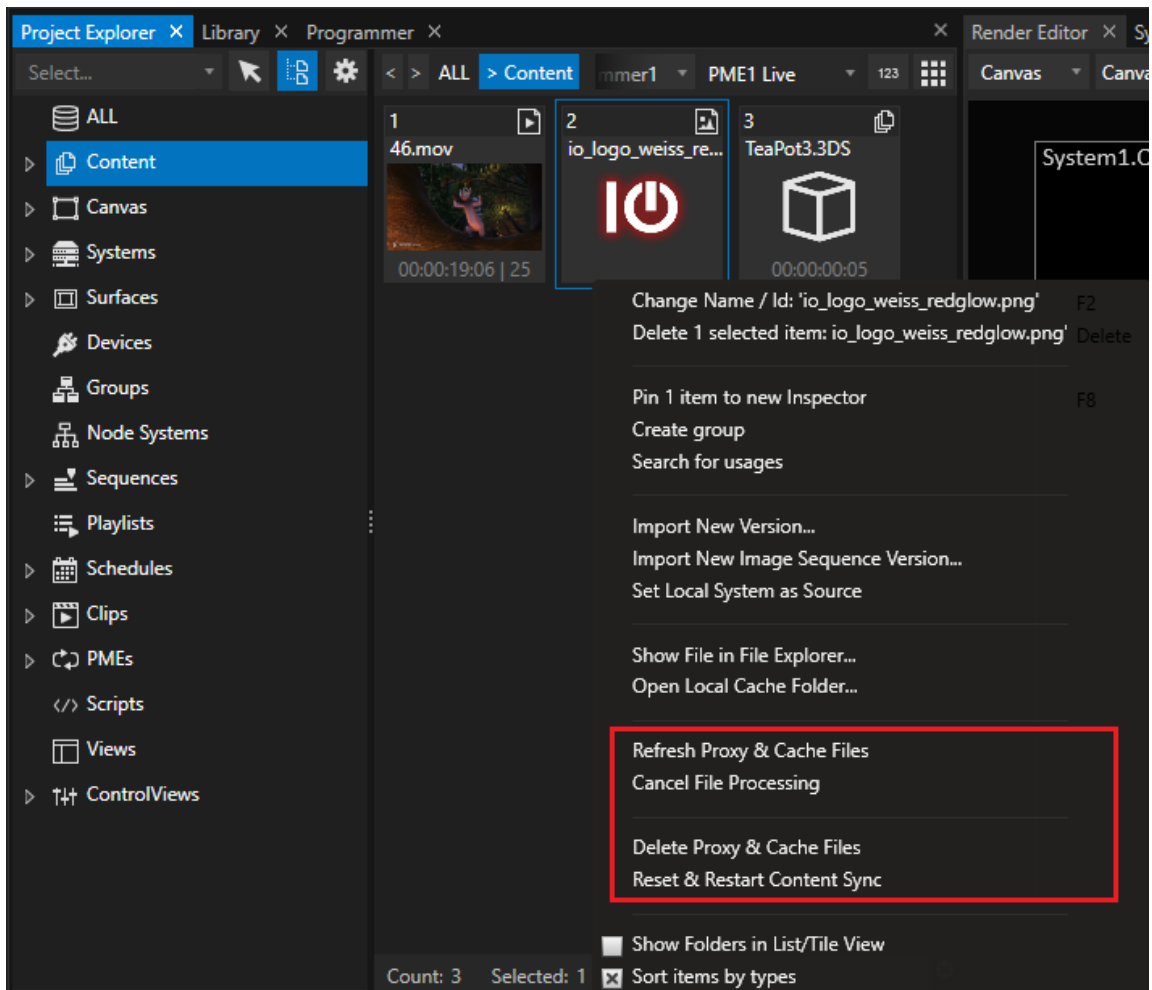


*The content transfer monitor gives you information about the transfer progress for each single content in your project.*

*When the window is empty all transfers have been performed or not yet started.*

## Restart Content Sync





If something went wrong and Content was not distributed correctly, you are able to **trigger another content distribution**.

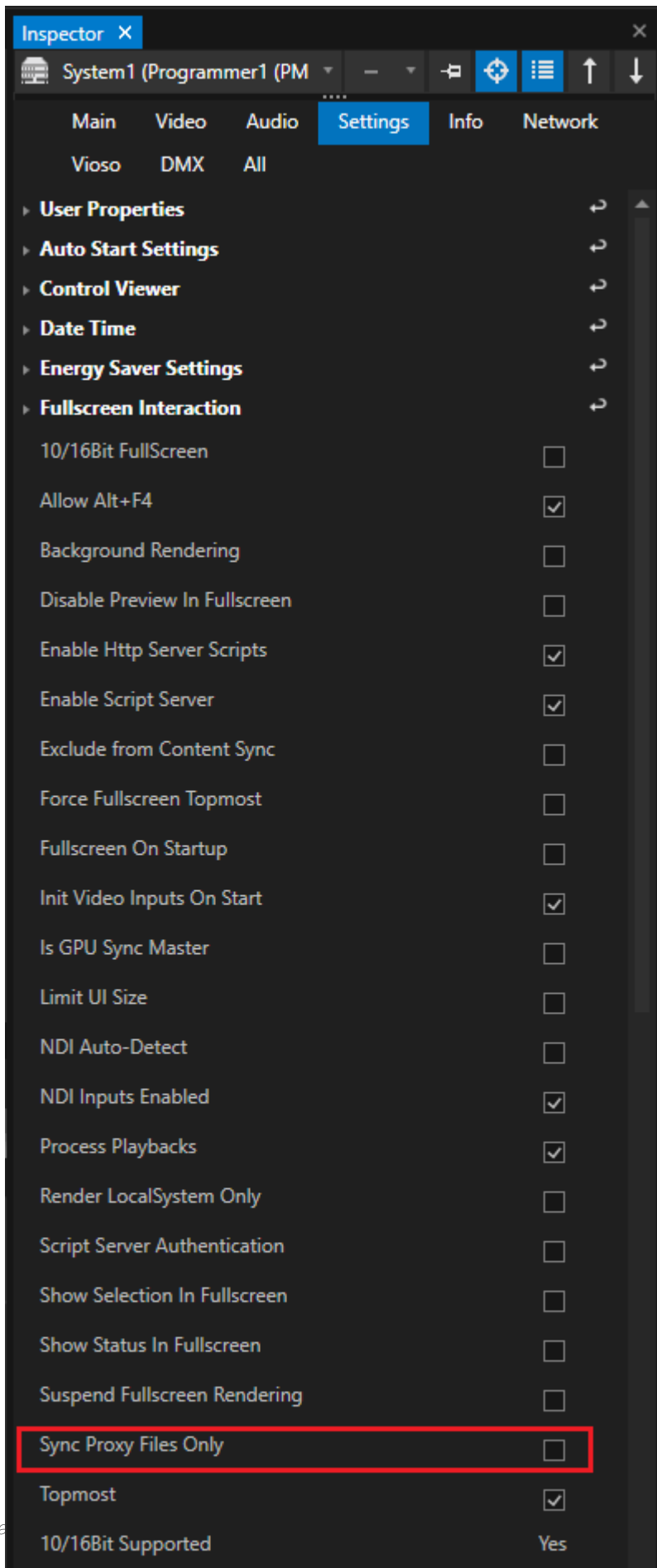
- Go to **Project Explorer**
- Select your **Content** there and open the **context menu** with a right-click
- Select "**Reset & Restart Content Sync**"

There are also options to recalculate the Proxy and/or cache Files (**Refresh Proxy & Cache Files**) or to delete them

After a refresh is triggered, the encoding will start again and the new files should be distributed by Content Sync

## Settings

For a System



**Sync Proxy files only.**

*When set, only Proxy Files are synced with this System. The original file will be excluded*

**Content Sync Adapter. ( Network Tab in the Inspector)**

*Define a network adapter on which the content sync should run.*

*Use this function to separate the Content-Sync from the rest of the network data. Useful for e.g. large amounts of data.*

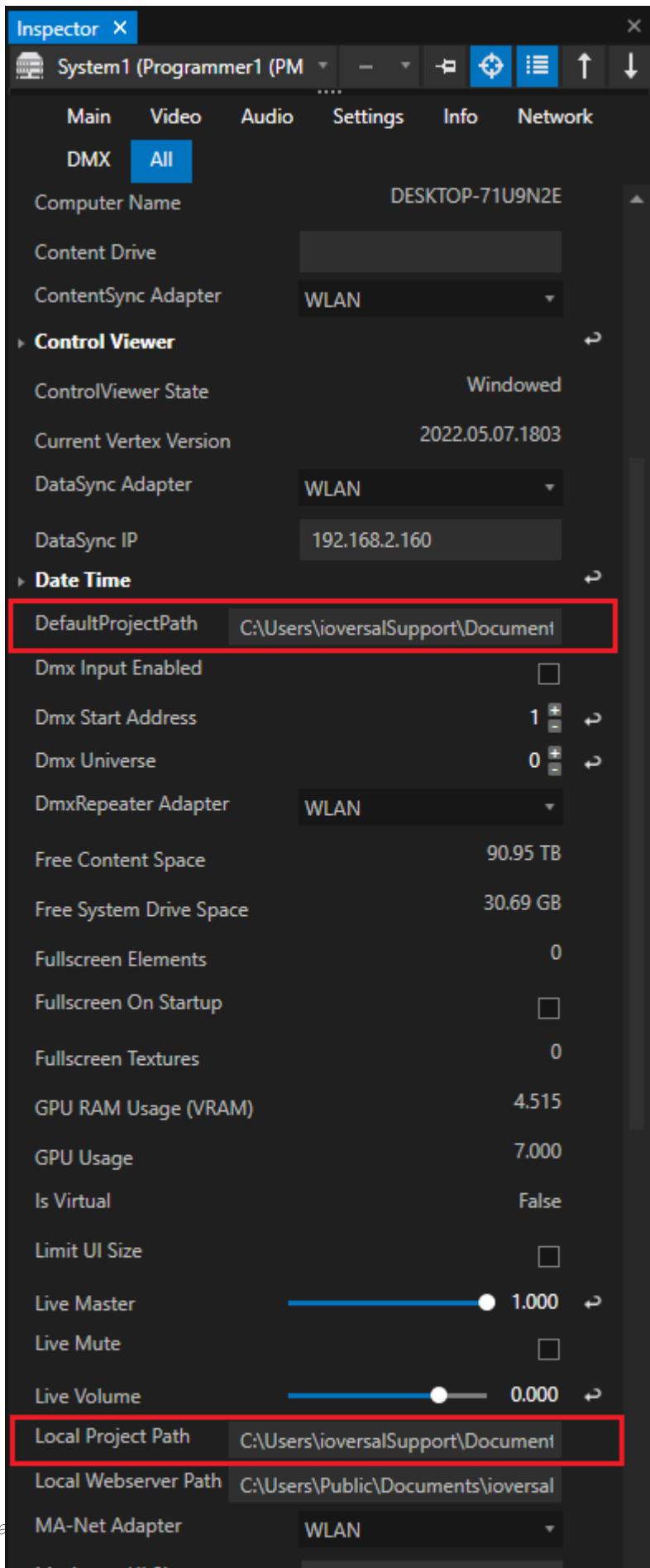
**Local Project Path**

*Defines the folder in which all Content data is written on as System*

**Default Project Path**

*Path that was set in the Installer first. Default Path that is taken into account as Local project Path*

*Change to permanently define a different target folder or drive for all project files*



## Project Settings

*Global Settings in the Project Settings*

### **Auto Content Sync**

*Default: enabled*

*Disable to deactivate Content Sync works automatically in Background*

### **New System Set Proxy only**

*default: Disabled*

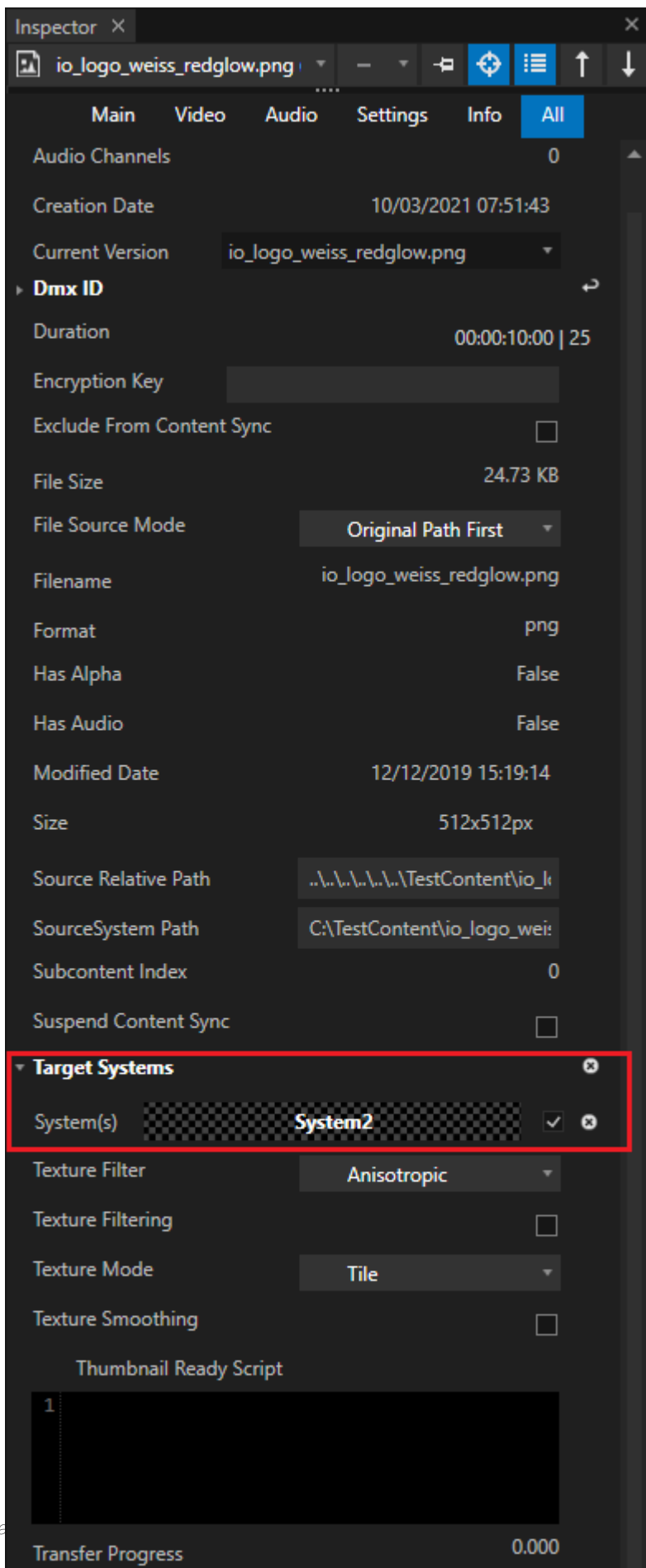
*When enabled, any new System added to the project will only receive Proxy files and not the original files.*

## Target Systems

*With the **Target System Property of a Content** you are able to define to which Systems this Content should be distributed.*

*By default, this Property is not set and empty: All Content will be distributed to all Systems.*

***Drag a System** from Project Explorer **to the Property field into Inspector** to set this System as Target System*

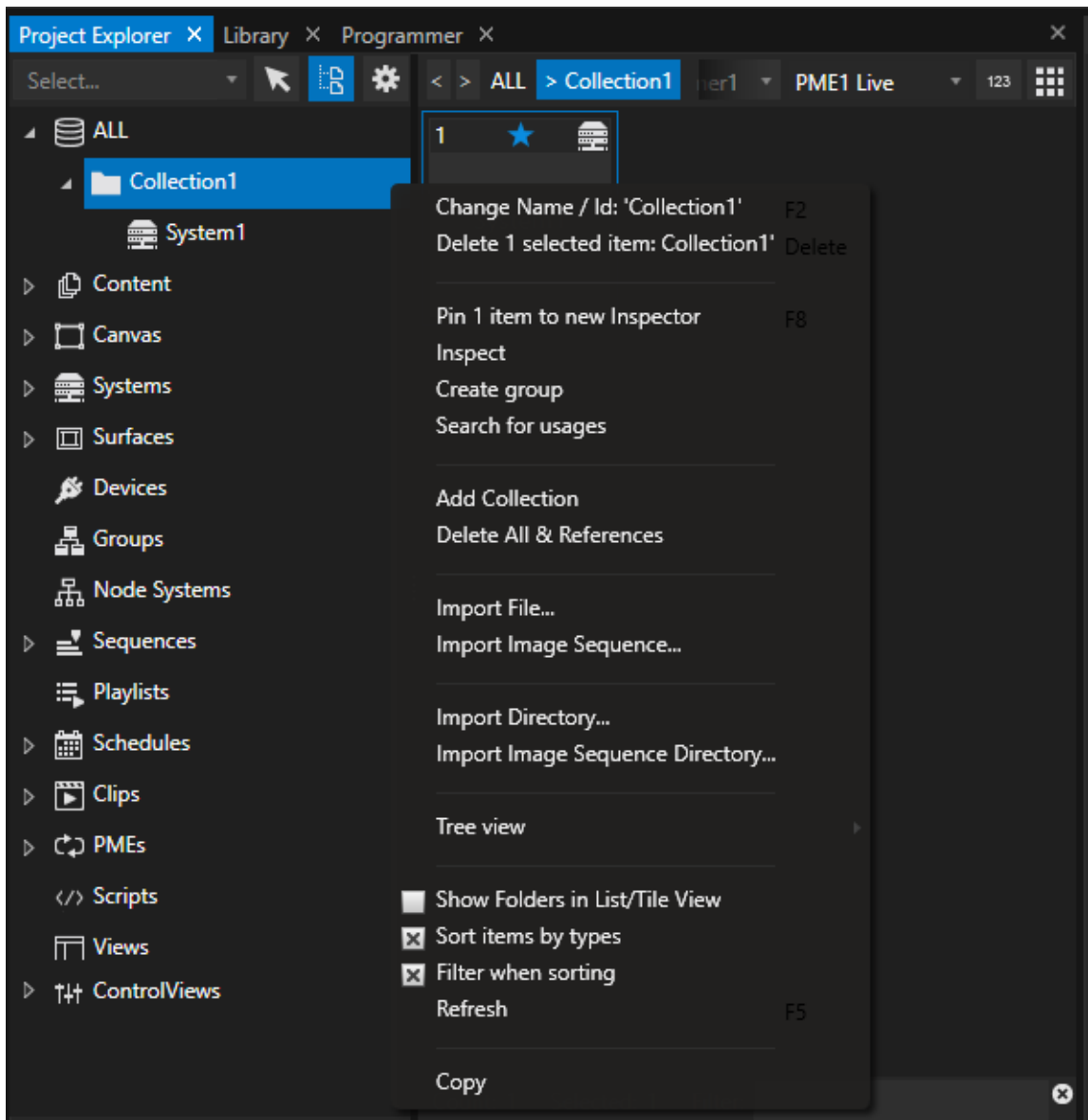


## Presets for Target Systems

There are tools to automatically set Target Systems on Content Import

### Collection in "All Section"

- Add a new Collection for the "All" Section into Project Explorer
- Drag System into this Collection
- Right-click on the collection and import Content
- The Systems into this Collection are set as Target Systems for all Content you import on this way.





### **Import Content on a System**

- Right-click on a System into project Explorer and open Context Menu to import new Content
- This System is set as Target System for the imported Content.

## 5.6.5 Data Routing

- The **mesh architecture of VERTEX** includes a **data routing** which **shares Device Data and Control Protocols over all Systems**
- Use the e.g the DMX-Routing Editor to **repeat and route Art-Net™, sACN or DMX-512** from one system to one or all other Systems into the Project

## Data Routing Options

With VERTEX handling of your Protocol and Device data will become easy.

Data that comes in on a System can be repeated and used on another System into your Project.

### Device Data and Protocols

**Device Data** that is caught on a System could be used on another System.

Use the Data of a Device which is connected to a System 1 on another System 2.

The **type of the data and the incoming protocol depends on the Device type** into the [Library](#)

[Learn more about Devices](#)

### Manage, Route and Transform Control Protocols like DMX-512, Art-Net™ or sACN

**Route and Repeat incoming Lighting Control Protocols.**

Input e.g. DMX-512 with a ioversal DMX IO on System 1 and output this Data as sACN on System 2

[The DMX Routing Editor](#) gives you the full flexibility

## 5.6.6 Backup

- There's **2 options for backup** in VERTEX: **partial backup** and **redundant backup**.
- Switching between the modes can be done **via UI** or triggered with a **script command**.
- Default backup mode is partial - the **mode can be set in the project settings**.

# Backup Mode

## Partial

A **backup system** is a **session member** that takes over the **role of another session member** or the **master**.

The **master role** can be taken over **from every system** in your project.

A system can **take over the render part** of another system.

VERTEX offers a great flexibility managing the amount of backup systems per project.

## Redundant

Your live-setup is **completely mirrored on a backup group**:

A group of live systems for instance could consist of one master and two session member systems.

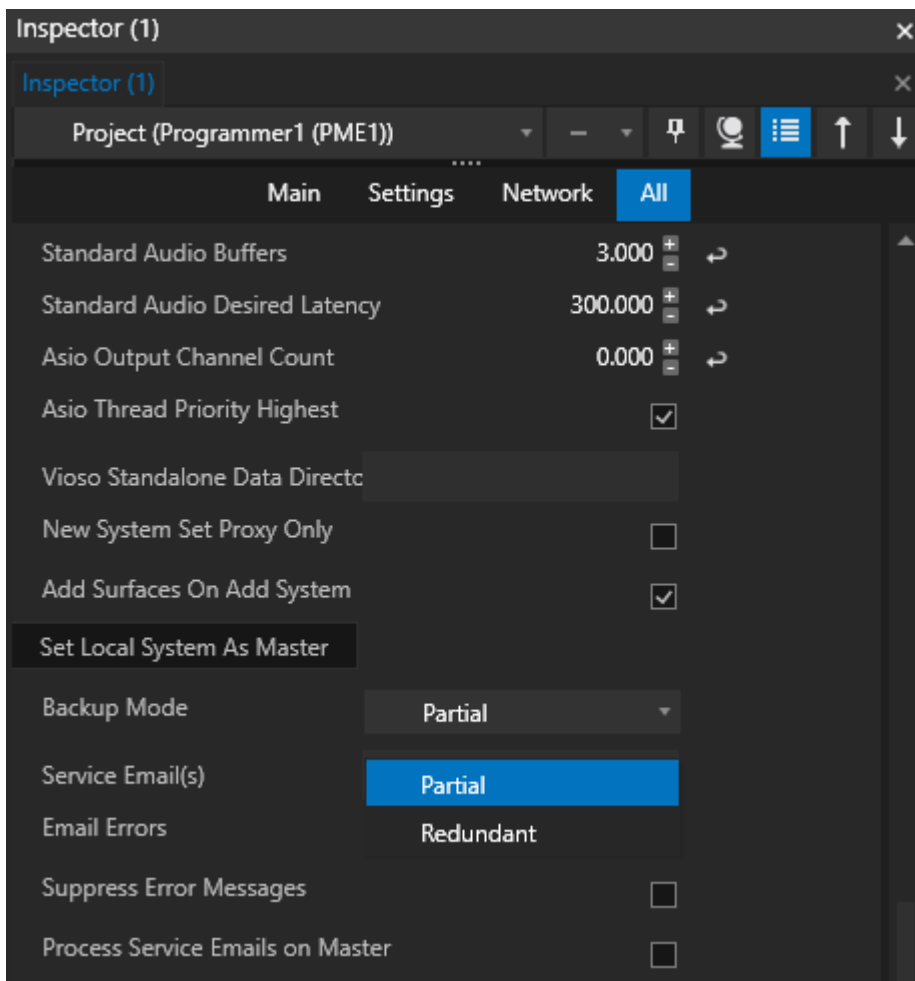
In a redundant setting, a mirrored group of one backup master and two backup session members is held in reserve, ready to take over any time.

## Set Backup Mode

The **default mode** for backups is **Partial**

The mode **can be set** for a project in the [Project Settings](#).

Depending on the selected mode, the menu entries for Backup in the main menu will change.



## Partial Backup

### Master

Every system in your project - including a session member that is kept in case a backup is needed - can take over the master role.

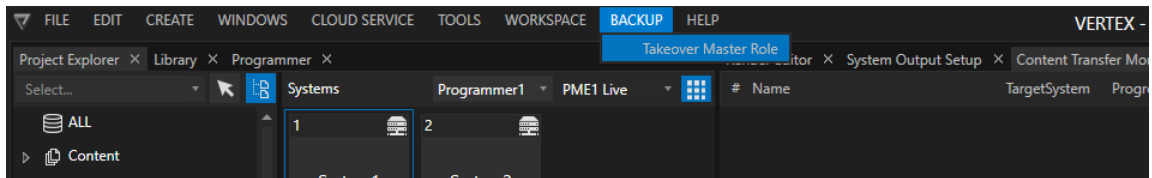
How to switch:

#### 1. Via the UI

Go to the **Session Member System** that should take over the master role,

Go to **Backup tab** of the **Main Menu** there and select "**Takeover Master Role**"

The **Session Member** switched its role to **Master**. Now the former master (if still alive) will become a **Session Member**.



## 2. Use a Script Command

```
SystemsManager.BackupTakeOverMasterRole
```

This script command can only be triggered **on the local system** and also externally via [VERTEX API](#).



### Sync-Clock Source System

Switching the Master Role will not change the Sync Clock Source System of a project.

If the former Master is defined as Sync Clock Source System and still alive, it keeps this function even after a master switch.

If the former Master is dead for some reason, the Backup Sync Clock Source System that is defined in the Project Settings, will take over this function.

Please consider this, if you want to change also the Sync Clock Source System on a master switch. You can e.g. do this with an additional script command in a backup script.

## Session Member

Your backup system is already a session member in your current project - there is no role change needed as for master.

Each VERTEX system has a property in advanced mode called **Render System** and **Audio System**.

In a backup case for one session member, another session member can **simply take over the tasks for rendering or audio playback**.

Condition of course: it has the **same number of outputs** with the **same resolution**. The backup system must have the **same numbers of surfaces** with the same render target size.

### Example:

System2 should render content for System1

Select System1 in the inspector.

Change the **Render System** property by dragging System2 from the project explorer and dropping it into the property field of the inspector.

Now System2 takes over all render tasks from System1.

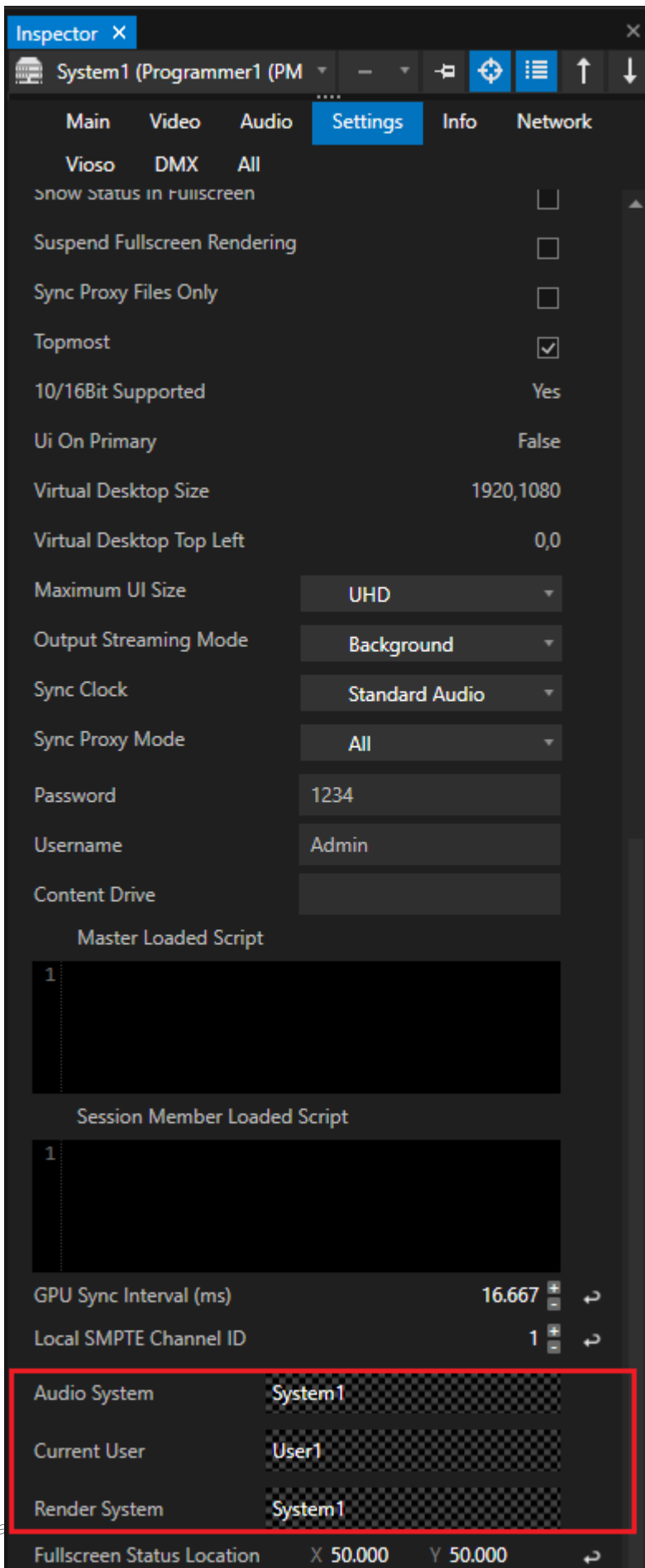
The same works for audio systems.

How to switch:

### 1. Via the UI

In the inspector select the faulty system that needs to be backed up.

Drag a new system from your project explorer and drop it into the render system property.



## 2. Script Commands

```
// Set System2 as Render and Audio Backup for System1
System1.Settings.RenderSystem.Value = System2
System1.Settings.AudioSystem.Value = System2
```

# Redundant Backup

## Backup Group

### **a live group consists of**

- live master
- number of live session members

### **a backup group consist of**

- backup master
- number of backup session members

### **Backup groups can have one of the following two states:**

#### 1. **Connected:**

*All changes on the live system are pushed to the backup system.*

*When connected, the project state from backup master will be overwritten with the project state from live master.*

*Use to update your backup group to the newest project changes. The project state on backup master is overwritten and updated.*

*When connected, the live master also acts as master for all members of the backup group.*

#### 2. **Disconnected:**

*If the backup group is disconnected, both groups - live and backup - act autonomously.*

*The live master controls all session members of the live group.*

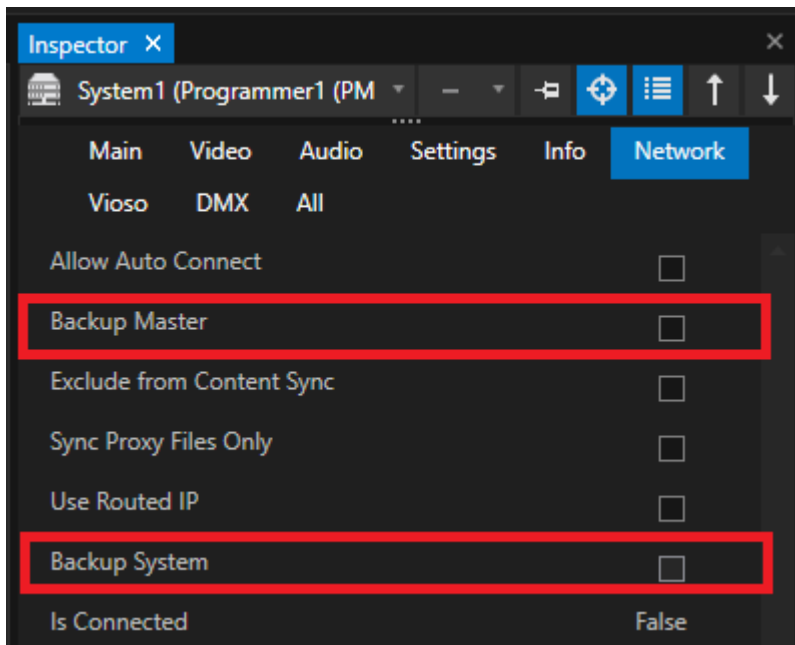
*The backup master controls all session members of the backup group.*

*Changes on the live group will not be pushed to the backup group.*

*The project state of the backup group corresponds to the time when the groups were separated.*

## Define Members for Backup Group

1. **Add all systems as session members** to your Project.
2. **Switch Backup Mode** in your project settings to **"Redundant"**.
3. **Select Systems in the inspector** and enable the following:
  - Backup Master:** if enabled, this system will become master of the backup group.
  - Backup System:** enable Backup System for all Session Members of the Backup Group



Please take care, that all members of a Backup Group have assigned the **Content** they need to play back when sharing content only to certain **Target Systems**, please check your System and content settings into Inspector

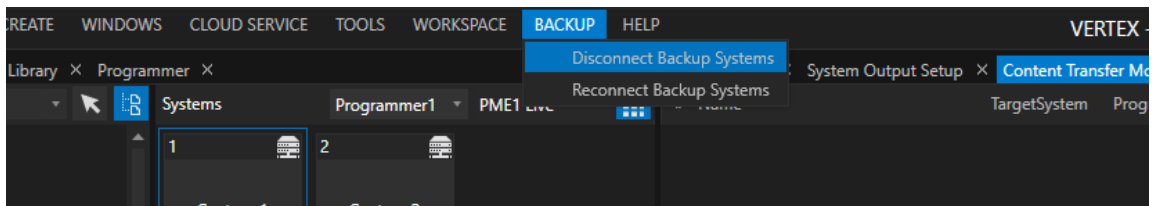
## Connect and Disconnect Backup Group

### 1. Via User Interface

Go to Backup Tab into Main Menu

Disconnect or Reconnect backup Group





## 2. Script Command

```
SystemsManager.BackupGroupReconnect  
Systemmanager.BackupGroupSeparate
```

This Script Command could only be triggered **on the local System** - of course also from external over the [VERTEX API](#)

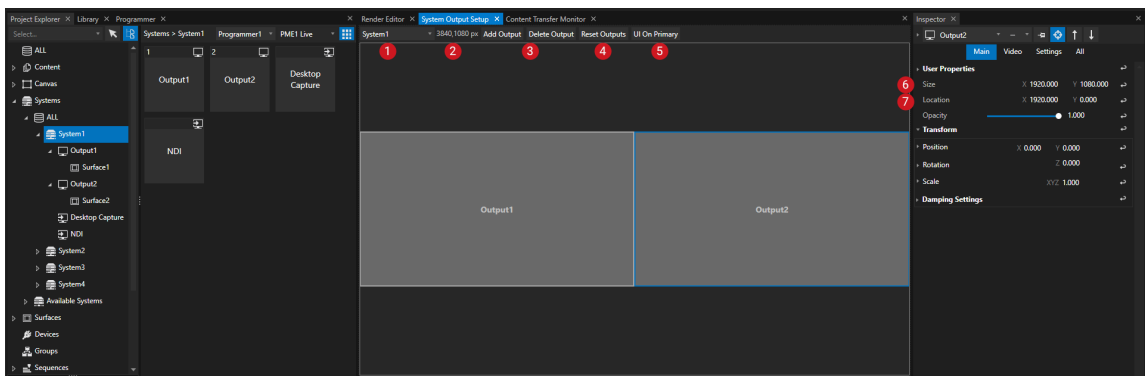
## 5.6.7 Advanced Settings

The documentation for the advanced feature-set of VERTEX is in progress and will be updated step by step  
Until then: Please drop us an E-Mail with your "How-to-do-this-in-VERTEX" question to [support@ioversal.com](mailto:support@ioversal.com)

## 5.7 System Output Setup

- The **System Output Setup** is the editor where you **manage the outputs of a system**.
- **"UI on Primary"** is a quick setting for a **mixed usage of UI and fullscreen** on the same system.
- **Reset Outputs** will reset all output settings of a system back to current Windows 10 desktop settings.

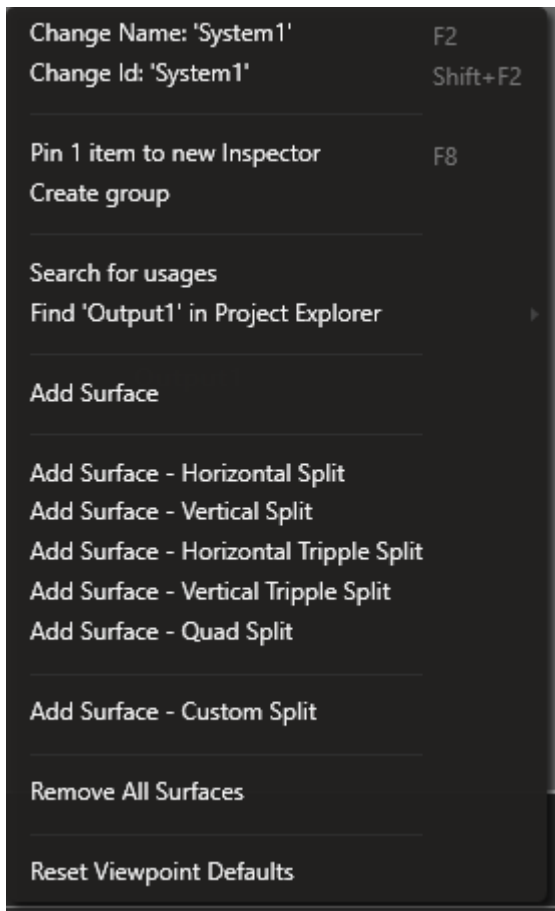
### User Interface



1	<b>Select System</b>	<b>default:</b> local system <i>In a project with multiple systems you can manage the output settings for each session member system like so: select the system you want to assign outputs to. Manage all system-output configurations from the same editor.</i>
2	<b>Desktop Size of a System</b>	<i>Shows the total Windows 10 desktop size of your system. VERTEX takes over the size that is set into your Windows 10 desktop management.</i>
3	<b>Add or Delete Output</b>	<i>Adds a new output or deletes a selected output.</i>
4	<b>Reset Outputs</b>	<b>Resets outputs of a system.</b> <i>VERTEX takes over the desktop settings from Windows 10 again. Use when outputs have changed since last project file load or just to take over again the Windows 10 settings.</i>

		<i>Old outputs are deleted, new ones are created for this System. For surfaces that were assigned only the "old" outputs the output reference is removed, but the surface not. After resets new surfaces are created for each output. If want to use the old ones again, reassigned the to the outputs.</i>
<b>5</b>	<b>UI on Primary</b>	<p><b>Sets VERTEX User Interface on Screen that is defined as primary Screen in Windows 10 Desktop Setup.</b></p> <p><i>Use when you want to work with the Vertex UI on one screen and fullscreen renderer on the other connected screens.</i></p> <p><i>"UI on primary" removes the primary output from system output setup.</i></p>
<i>Select an output with your mouse - properties are shown in the inspector</i>		
<b>6</b>	<b>Size</b>	<b><i>pixel size of the output</i></b>
<b>7</b>	<b>Location</b>	<i>location of the output into the Windows 10 desktop window</i>

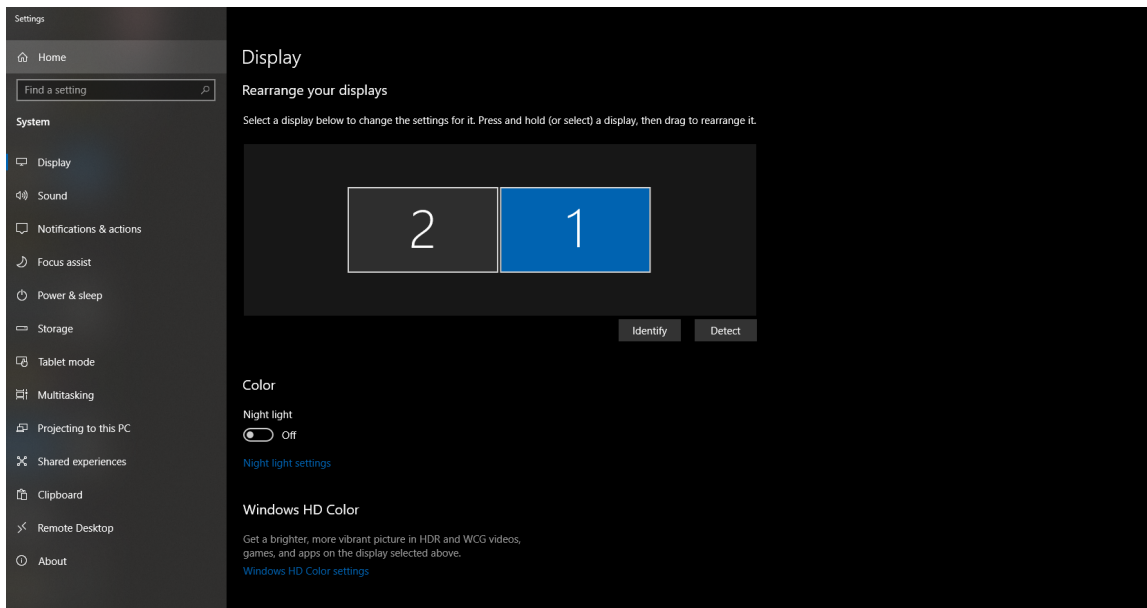
## Context Menu



Right-click on an output to **open the context menu**.

Here you will find some **quick actions to create and assign one or multiple surfaces**, i.e. a quad-split for a 4 output-vertical NVidia MOSAIC or AMD Eyefinity Setup.

## Windows 10 Desktop Settings



VERTEX appropriates the following properties from the Windows 10 desktop settings:

- the number of connected screens,
- total Windows 10 desktop size,
- the primary screen and output arrangement.

Check these settings first. The screen's number in VERTEX can differ from Microsoft Windows depending on your setup and assignment.



#### **DPI scaling has to be set to 100%**

Please double check the scaling for your screens in Windows 10 desktop setup. We strictly recommend a 100% scaling for each of your screens. Especially for high resolution screens Windows 10 automatically sets the scaling to 125%, 150% or even higher. This can cause side-effects with calculation in the render editors. To avoid trouble, use the advanced scaling settings of Windows 10.

## 5.8 Playback Live and/or in Preview

*Learn the basics about Live Playback*

### Playback Mixing Engines (PME)

*A PME hosts all Playback and could be Live or Preview.*

*Mix Levels of a PME defines the final signal that is going out to live*

*Create as many Preview PMEs as you need.*

### Preview

*Work in Live and in Preview - Switch Editors between PME Live and Preview*

*Build workspaces that combine Live playback and Preview editing.*

### Fullscreen Renderer

*Enter or Leave Fullscreen*

*Get information about the Fullscreen options of a System*

### Output Stream

*Stream your rendered Content as an NDI Stream out of VERTEX*

### Audio Playback

### 5.8.1 Playback Mixing Engine

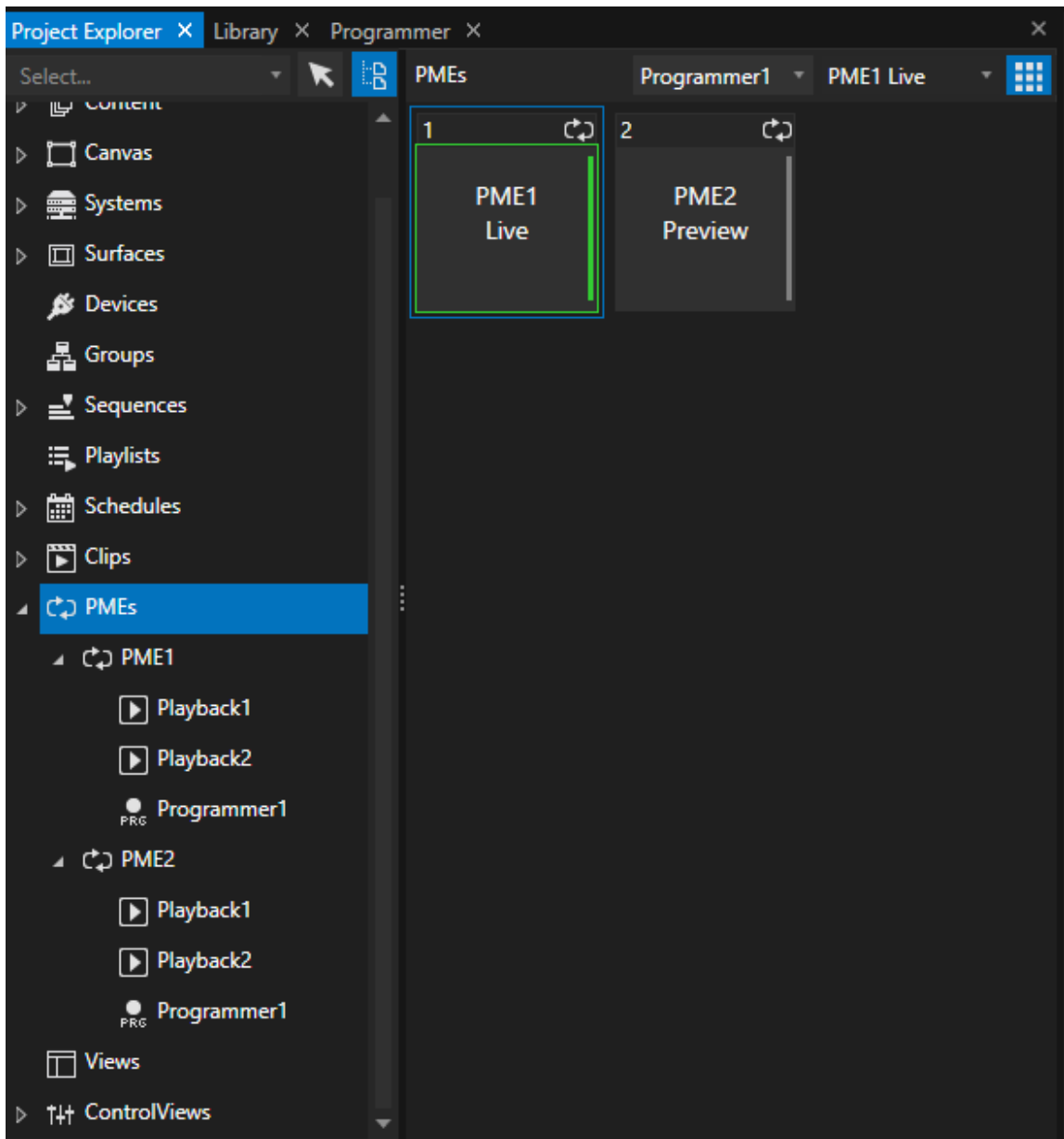
- *PMEs into VERTEX **host all Playbacks***
- *by default **there are 2 PMEs** - one **for Live** and a second **for a Preview***
- *The PMEs into VERTEX are **like Video Master Faders** that define if the final Signal goes out to Live or Preview*
- ***For every Render Editor and Playback Editor** you can select whether it **should show Live or (one of the) Previews***

## Playback Mixing Engines (PMEs)



PMEs are the **lowest level of the Playback Chain** into VERTEX. The **final Result from the PME goes out to Live or Preview Renderer.**

Each PME **hosts all Playbacks of a Project** - With the terms of a video mixer: PME's are your master groups or faders.



By default there are 2 *PMEs* into a *VERTEX* project:

- *Live* and
- *Preview*.

## Live

*Live PME* always goes out to *Fullscreen Renderer*.

Audio from all playbacks running into *PME Live* is going out the the Audio Device that was [set as live device](#)

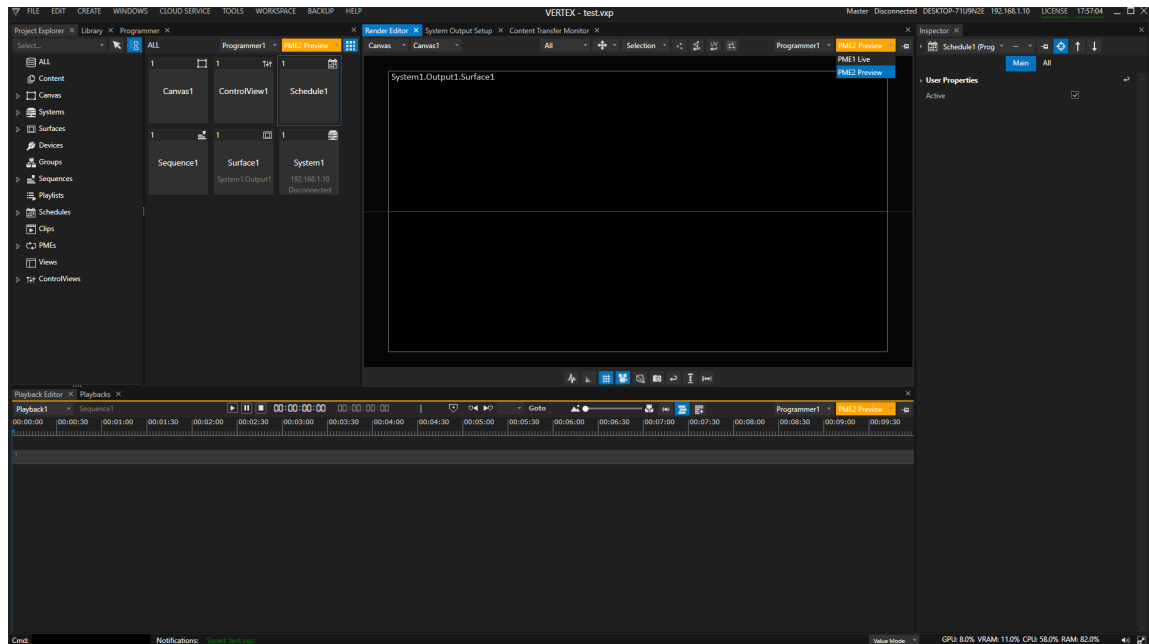


## Preview

A Preview PME renders a [Preview of your Playbacks](#).

**Into each Playback Editor and each Render Editor, you are able to choose which PME should be shown.**

Switch to e.g. PME 2 to work with the Preview



You are allowed to **create additional PMEs** - depending on how many different Previews you want to have for more. complex projects.

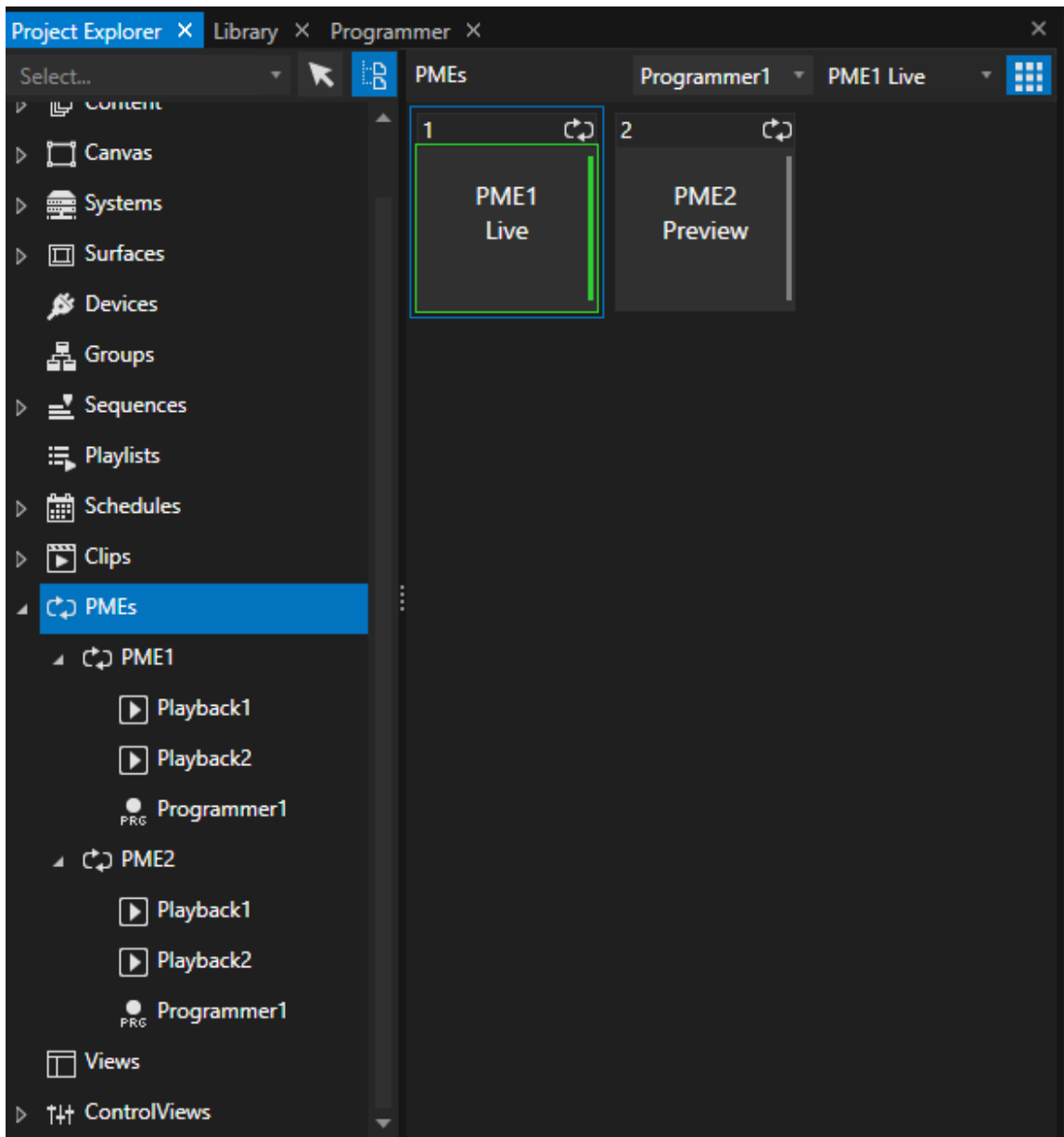
Regardless of the number of your Playback Mixing Engines, **only one PME could be the Live PME**

## Programmer and PME

The PMEs also hosts your Programmer(s).

Depending on the selected PME into your Playback Editor, your changes into Preview will be stored as Preview Changes into the Programmer.

## Mix Level



*Each PME has a Mixing Level - like a Video Mixer has.*

*The Mixing level decides about the signal that is sent to your live output. Value 1 is sent to Live, Value 0 not.*

*By default the **Mixing Level for Live** is set to 1.*

*The **mixing level** for all **Preview PMEs** is set to 0. The signal of a Preview PME is displayed into e.g. a Preview Render Editor, but the signal is not mixed to the live output.*

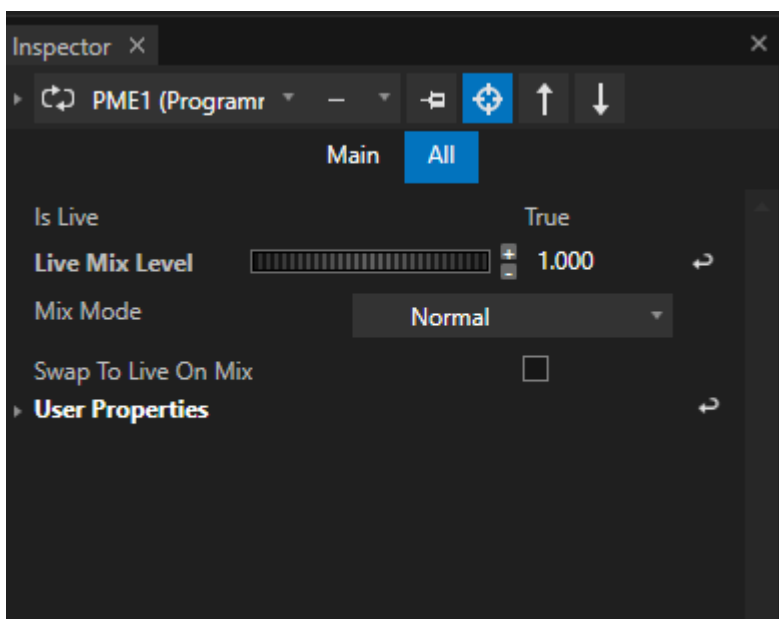
*By increasing the **mixing level** up to 1 you are able to mix preview into live*

*There is also the option to **swap preview and live***

## Create a new PME

- Go to Main Menu, open "Create" tab and select "Create PME"
- or
- Go to the PME section into Project Explorer, right-click to open the context menu and select "Create new"

## Settings



### Live Mix Level

*If set to 1, signal is sent to Live Output and Fullscreen Renderer*

*If set to 0, signal is not sent to Live Output and only shown into Preview Render Editors*

### Mix Mode

*Define the Mixmode when fading Live Mix Levels*

*Options: Normal, Crossfade, None*

## Swap to Live on Mix

When enabled a Preview PME is swapped from Preview to Live if Mixing Level reaches 1.

The "old" Live PME is swapped to Preview.

### 5.8.2 Preview

- The concept of [Playback Mixing Engines \(PME\)](#) allows you to work **with an infinite amount of previews**
- **Work live and in previews simultaneously** and **fade your previews to live at any time.**
- **Into the VERTEX Editors you are able to switch between the PMEs.** Or just open multiple editors of the same type to work on live and preview in parallel

## Playback Mixing Engines (PME)

PMEs are the lowest level in VERTEX that host all Playbacks and are responsible for a Live or a Preview Mix

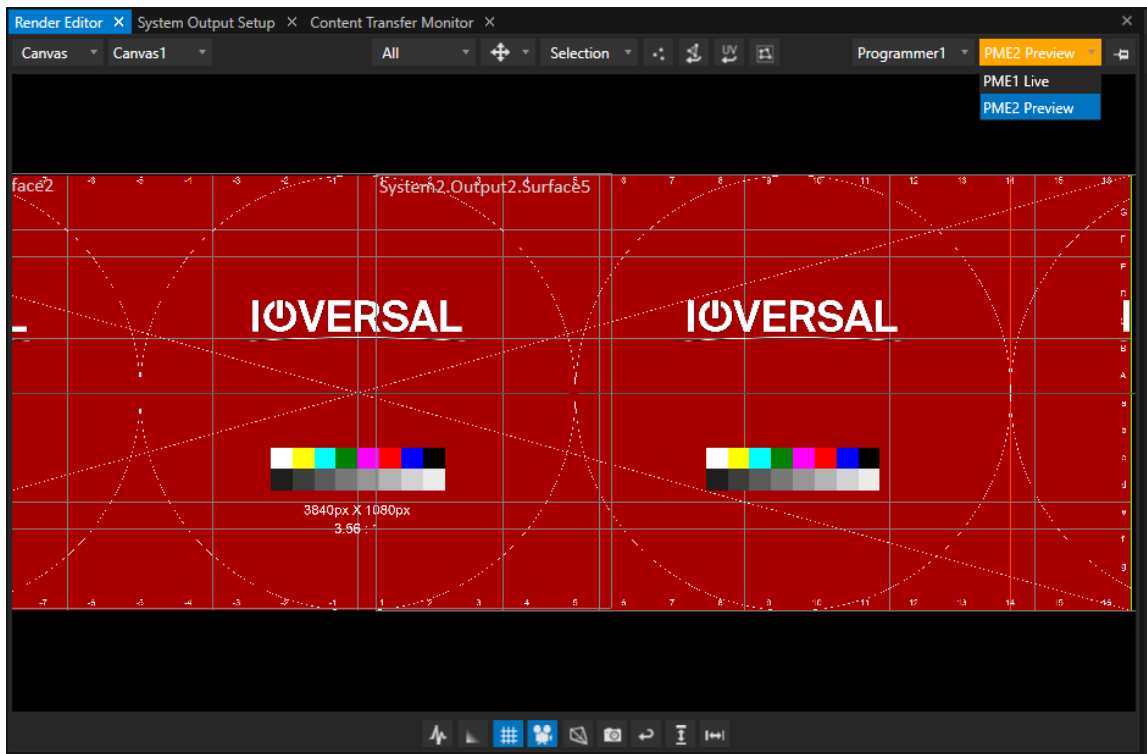
[Read more about PMEs and their Settings here](#)



The number of previews depends on your hardware resource. each render editor and PME takes hardware resources because a new render path is opened.

Total number of PMEs that could be rendered in parallel depends on your hardware setup. It is not limited by VERTEX but by your hardware..

## Switch between PMEs from Live to Preview



Render Editor is set to Preview PME

In most of the VERTEX editors you find a dropdown on top left with which you switch between Playback Mixing Engines



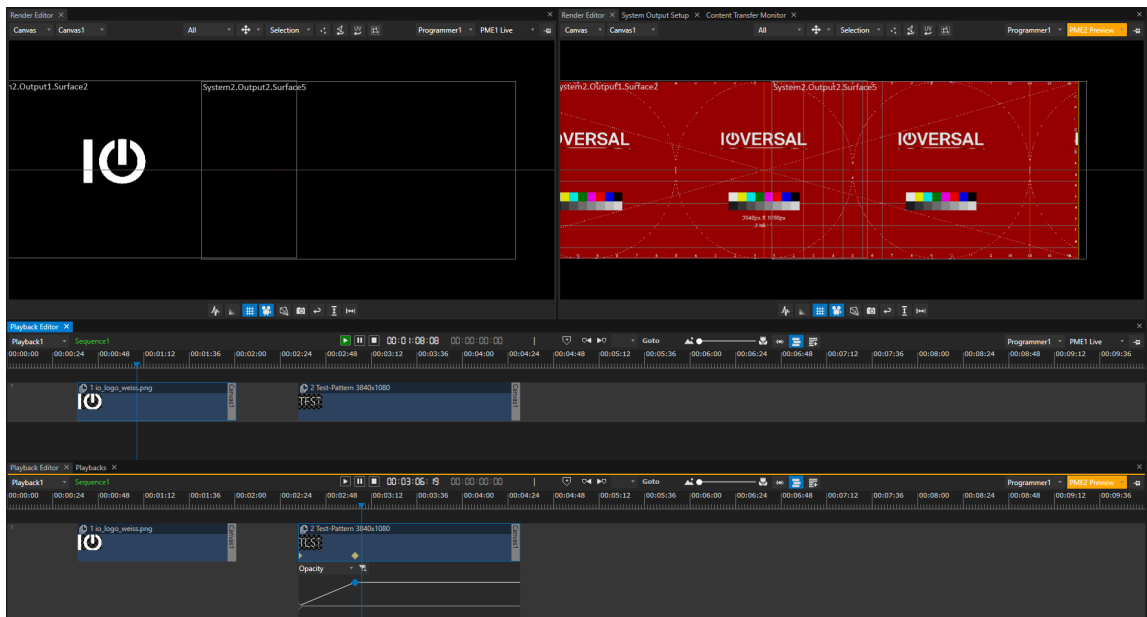
**Preview PMEs are marked in yellow** to give you visual feedback that you are working not in live but in preview.

## Work live and in preview simultaneously

Using the **modular window concept** of VERTEX you can build your **own workspaces** and save it as view.

These workspaces can **show live and preview simultaneously**.

You are able **to work in preview as well as live** - e.g. if you want to make changes during a rehearsal while live is playing.



Example for a Workspace:  
 2 Render Editors - one for PME Live, one is set to PME2 (Preview).  
 2 Playback Editors - one for PME Live, one is set to PME2 (Preview).

## Fade Preview to Live

The **mix level of a PME** the defines whether your content is played live or in preview.

[Read more about the Mixlevel and its settings](#)

### 5.8.3 Fullscreen Renderer

- Each Vertex system can run **in fullscreen mode**, in **UI mode**, or in **a combination of both on different screens**
- There are different ways to **enter the fullscreen mode: shortcuts, context menus, script or buttons**
- Make sure that no **render editor is working in the background**- this may **affect performance**.

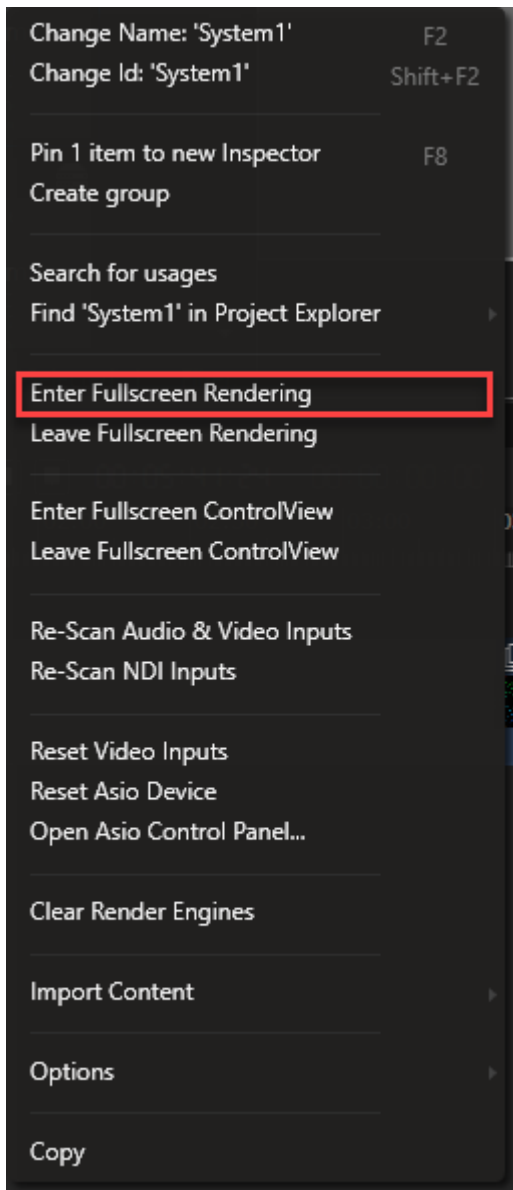
## Enter Fullscreen

**Fullscreen shows black and no content**

Check if you have added surfaces to your canvas and assigned them to the correct output.

There are **4 options** to enter the fullscreen renderer:

- Use **shortcut "CTRL + F"** on your local VERTEX system.
- Use the Fullscreen **button in the status bar** of your local system.
- Use the **context menu for a system** in the **project explorer** (also works remote for session member systems of a project)
- Use the **script command** `System1.EnterFullScreen`



## Combination Of Fullscreen And UI

Here is the workflow to view both the UI and fullscreen renderer simultaneously:

1. Disable the Topmost Setting on the local system's settings tab in the inspector. Now the Fullscreen Renderer can operate in the background.
2. Enter fullscreen mode.
3. On your keyboard, hit the shortcut ALT+TAB to toggle between applications and windows. Select VERTEX user interface from here.
4. If you have two outputs on your system, you might want to set up the surface-output constellation so that one output displays the rendered content and the other output is free for the UI.





**Make sure that the render editor is disabled or not opened**

Check settings (Inspector - Settings tab for a System) or manually close the render editor. If the render editor still renders in the background while in fullscreen mode, there will be a performance drop.

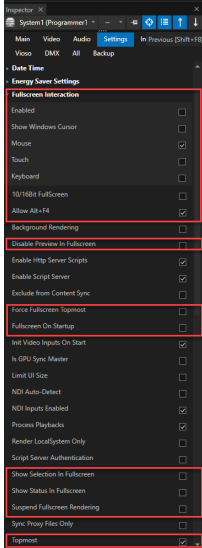
## Leave Fullscreen

Leaving fullscreen is possible with:

- The **shortcut CTRL + F** on the local system
- Use the **context menu onto a system** into project explorer (also works for session member systems into a project)
- Use **script command System1.LeaveFullScreen**
- If the use of shortcut **Alt+F4** is enabled in the system settings, use it just once to leave fullscreen mode. Using the shortcut twice will first close the render editor and then the VERTEX Application without asking you to save your project.

## Fullscreen settings for a system

<b>Fullscreen Interaction</b>	Enables input devices (mouse, keyboard, touch) in fullscreen mode for i.e. click scripts or URL content in interactive displays.
<b>10/16bit FullScreen</b>	Sets the color bit depth for fullscreen rendering. If you need to run your GPU in a higher resolution than 8bit, please check this setting before instantiating fullscreen mode for the first time. In case you need to change this setting in the middle of a session, VERTEX needs to be restarted.

	<p><b>Allow ALT+F4</b></p>	<p>Default: enabled - allows shortcut Alt+F4 for i.e. VNC Clients Pressing once without enabled fullscreen renderer will close the VERTEX application without asking to save your work. Pressing once while in fullscreen will first close the fullscreen renderer. Pressing the shortcut a second time will close VERTEX. Disable to prevent from pressing by mistake.</p>
	<p><b>Disable Preview in Fullscreen</b></p>	<p>Default: disabled With this option, the render editor will be disabled when fullscreen is entered. Does not work for a single master system - only for session members. Render editors will be enabled again when leaving the fullscreen renderer.</p>
	<p><b>Force Fullscreen Topmost</b></p>	<p>When enabled, this continuously ensures that fullscreen window stays topmost. Pop-up windows will be kept in the background.</p>
	<p><b>Fullscreen On Startup</b></p>	<p>When enabled, VERTEX always enters fullscreen on application startup.</p>
	<p><b>Show Selection in Fullscreen</b></p>	<p>When enabled, fullscreen renderer will show selected vertices or modifiers when editing surfaces.</p>
	<p><b>Show Status in Fullscreen</b></p>	<p>Shows overlay with status information (FPS, uptime, renderer frames) top left of the first screen.</p>
	<p><b>Suspend Fullscreen Rendering</b></p>	<p>Pauses fullscreen rendering temporarily.</p>
	<p><b>Topmost</b></p>	<p>When enabled the fullscreen window will be on top. Disable to <a href="#">view both UI and fullscreen</a> window simultaneously.</p>

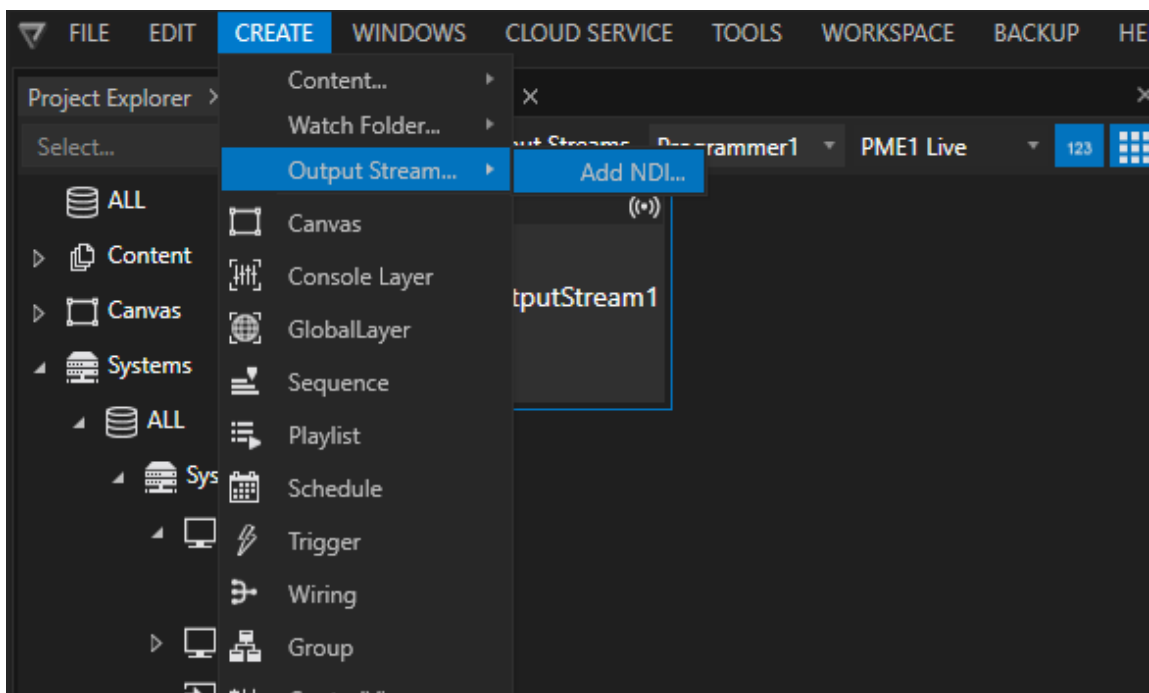
## 5.8.4 Output-Stream

- VERTEX is capable to **stream a Surface out as Output Stream** - currently only NDI is available
- Output-Streams could be **created with the Main Menu** ( Create) or with a **right-click on a Surface** in Project Explorer
- A **Fullscreen Renderer** has to run on your sender System or your **Sender System** has to be set into **Streaming Mode**

# Create an Output Stream

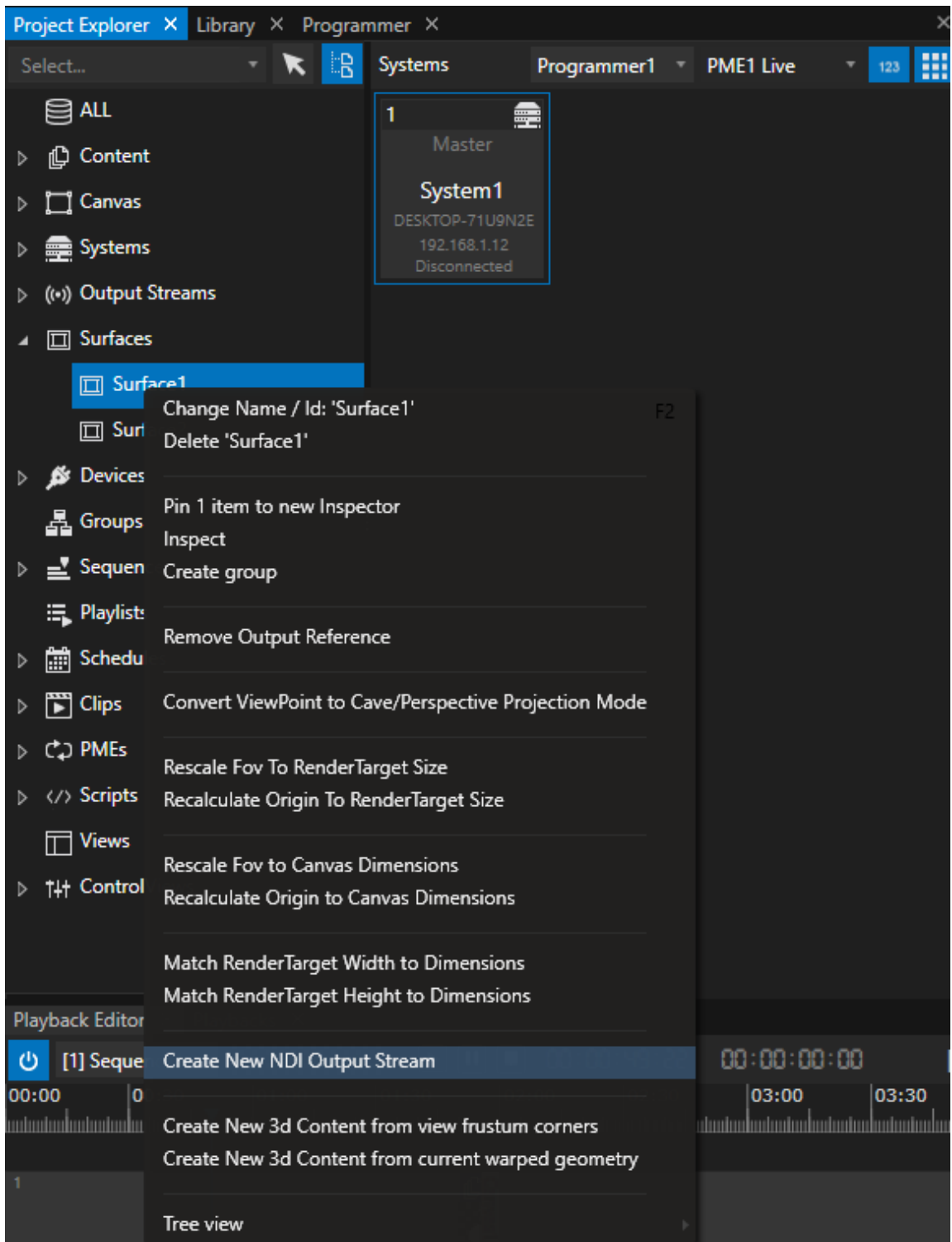
## 1. Empty Output Stream

- Go to the Main Menu, Open CREATE
- Select Output Stream
- add NDI Output Stream
- An Output Stream without an assigned Surface is created



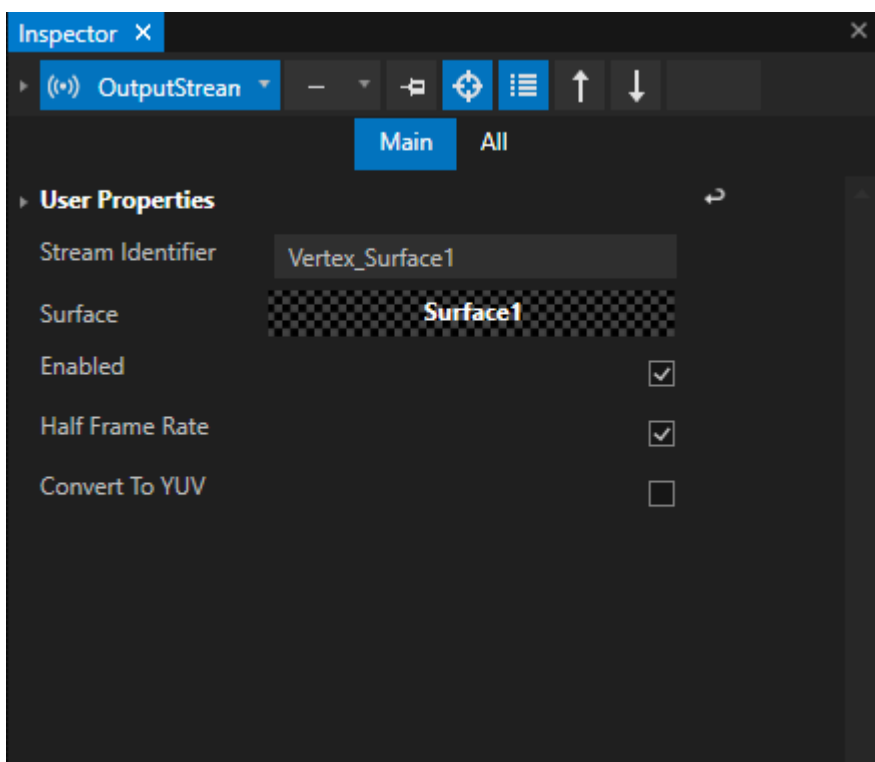
## 2. Output Stream based on a Surface

- *Go to the Surface section into the Project Explorer*
- *Right-Click on a Surface and open the Context Menu*
- *Select "Create new NDI Output Stream"*
- *A new NDI Output Stream is created and the Surface already is assigned to it.*

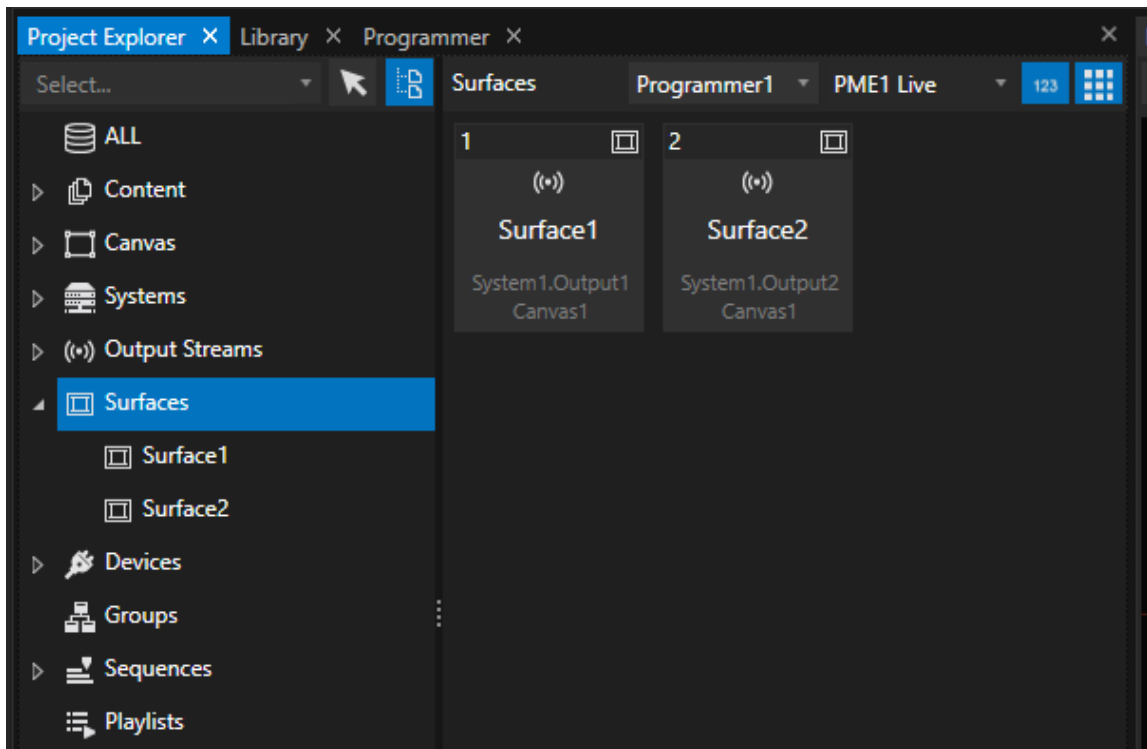


## Settings

- Open an Output Stream into **Inspector**
- To assign or change your **Surface**, drag a Surface from Project Explorer to the Property Field into the Inspector
- **Stream Identifier**: Changes the Name of your NDI Stream

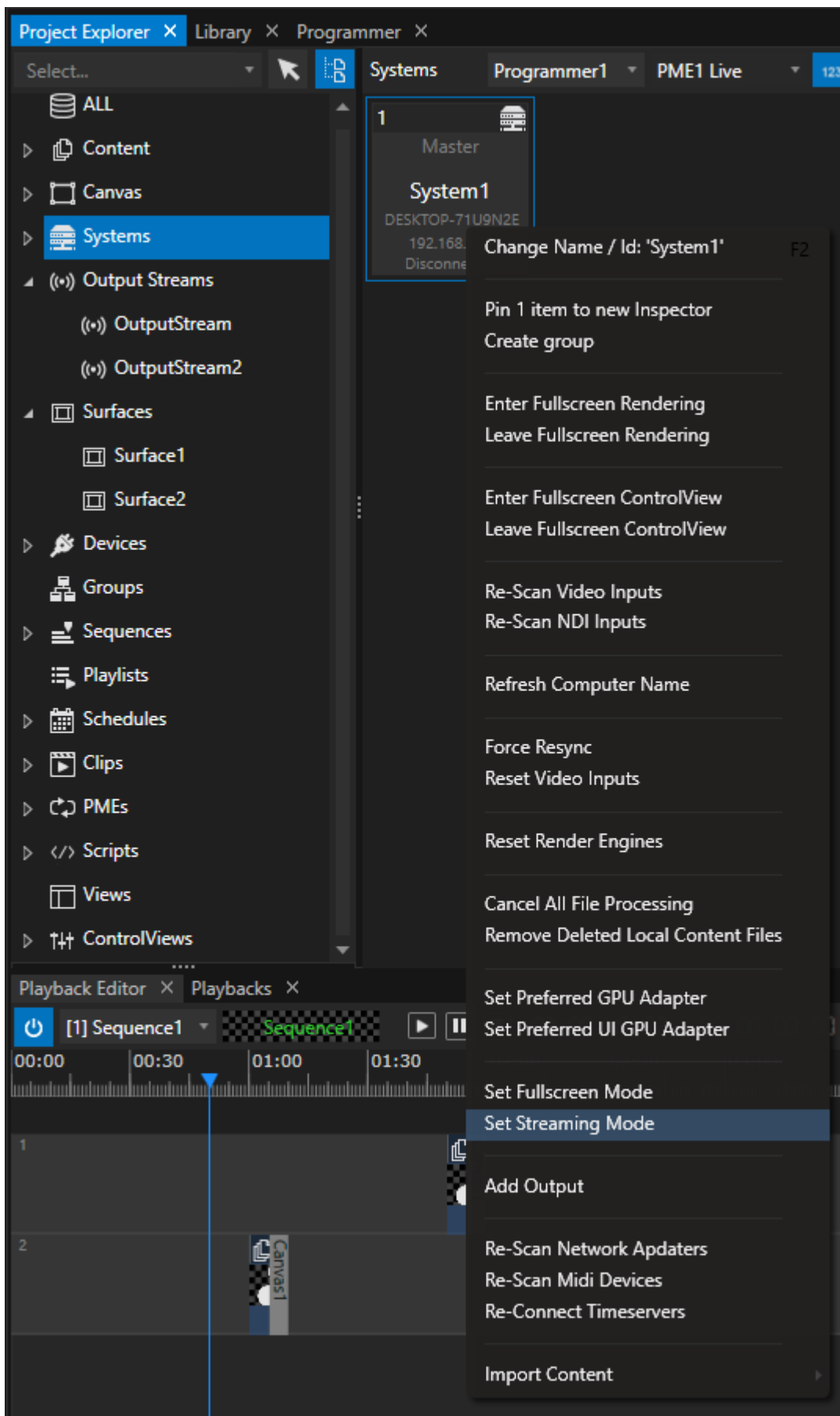


- **Surfaces** that are assigned to an **Output Stream** are marked into Project Explorer with a **sender symbol**:



## Send Fullscreen out as a Stream

- *technically, your content first **has to be rendered to send it as a stream.***
- VERTEX offers you **two options** to render a Stream in Realtime and send it out as NDI
- Right-Click on the System that should send your stream and choose between the 2 options below





### Set Streaming Mode

A hidden Fullscreen renderer is started in background. Streaming is possible without your local VERTEX application additionally has to a Fullscreen Renderer running.

### Set Fullscreen Mode (default)

No hidden Fullscreen Renderer is running in background. If using this mode, sending out a stream is only possible when you start the Fullscreen Renderer on this System

## 5.8.5 Audio Playback

- VERTEX differs between **Live Audio** and **Preview Audio**.
- You can assign a **separate audio device** for live and preview Audio
- You can **define a System** that should render Live Audio **for a Canvas or on System level**.

[Learn more about Audio Output Settings](#)



Please keep in mind that playing live audio is resource intensive.

Especially if you play many channels, the CPU will be heavily loaded. If you are using multiple systems in a project as session members, one option is to define a system for playing the audio tracks.

We recommend using an ASIO interface

## Preview and Live Audio

There is a main difference between **preview audio** and **live audio**:

### Preview Audio

- Preview audio plays out all audio from all playback mixing engines - That includes all playbacks in PME live and all playbacks in one (or maybe more) preview PMEs.
- Preview audio is played out from every system in your project.

### Live Audio

- Live audio plays out only audio from playbacks that are running in PME Live.
- Live audio is played out by an defined audio system that could be set for a canvas or a whole system.

By default live audio is played out for all canvas on the same system. It is possible to define a certain audio

system for a canvas.

## 5.8.6 Sync Clock

- The Sync Clock defines on which **clock source the frame playback is based on**.
- **Different options** for a clock source are: **System Clock, ASIO Audio, Wave Audio, SMPTE IO LTC, GPU Sync and Local Clock**
- Please also read the [Playback Sync-](#) chapter, when working in session mode.

## Settings

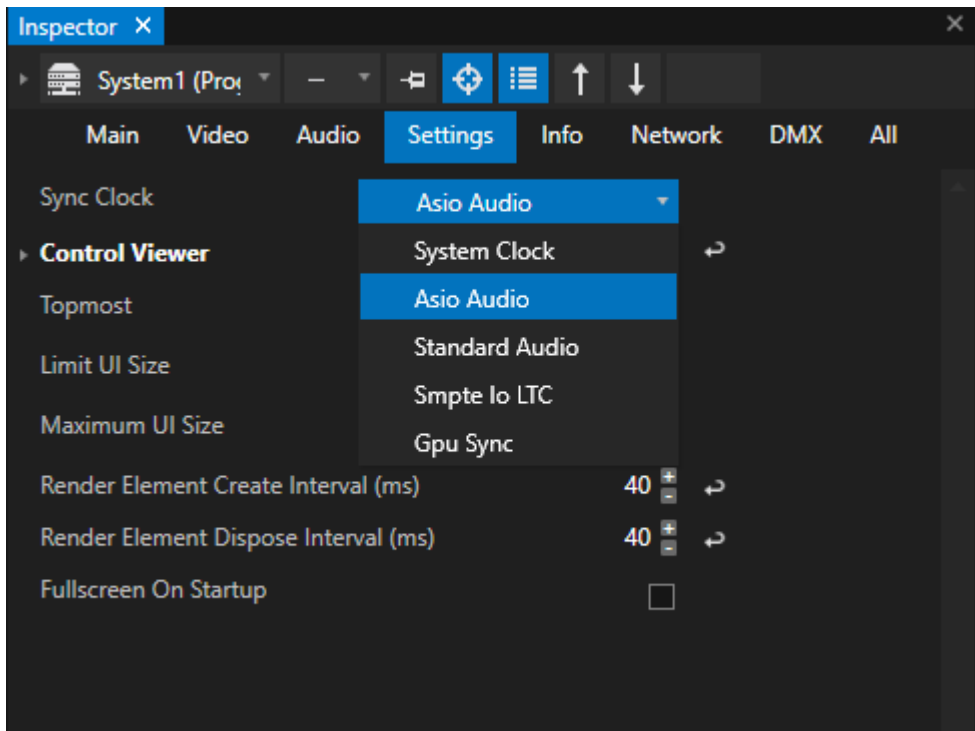
- Select your System and go to the Inspector
- Focus Settings tab there
- Select your preferred clock source in the Sync Clock dropdown list



**Top bar: green underlined clock indicates the system clock is synchronized correctly.**

This is especially important for a project [in session mode](#).

If the system clock is [underlined red](#), there is something wrong with Playback Sync and the Sync Clock



### **Changing Clock Source can affect the Playhead's position**

After selecting another clock source, the Playhead could jump to another position in the timeline.  
This only happens once directly after a switch of your clock source

## System Clock

Playback is locked on the VERTEX System Clock. Recommended when working without audio.

## ASIO Audio

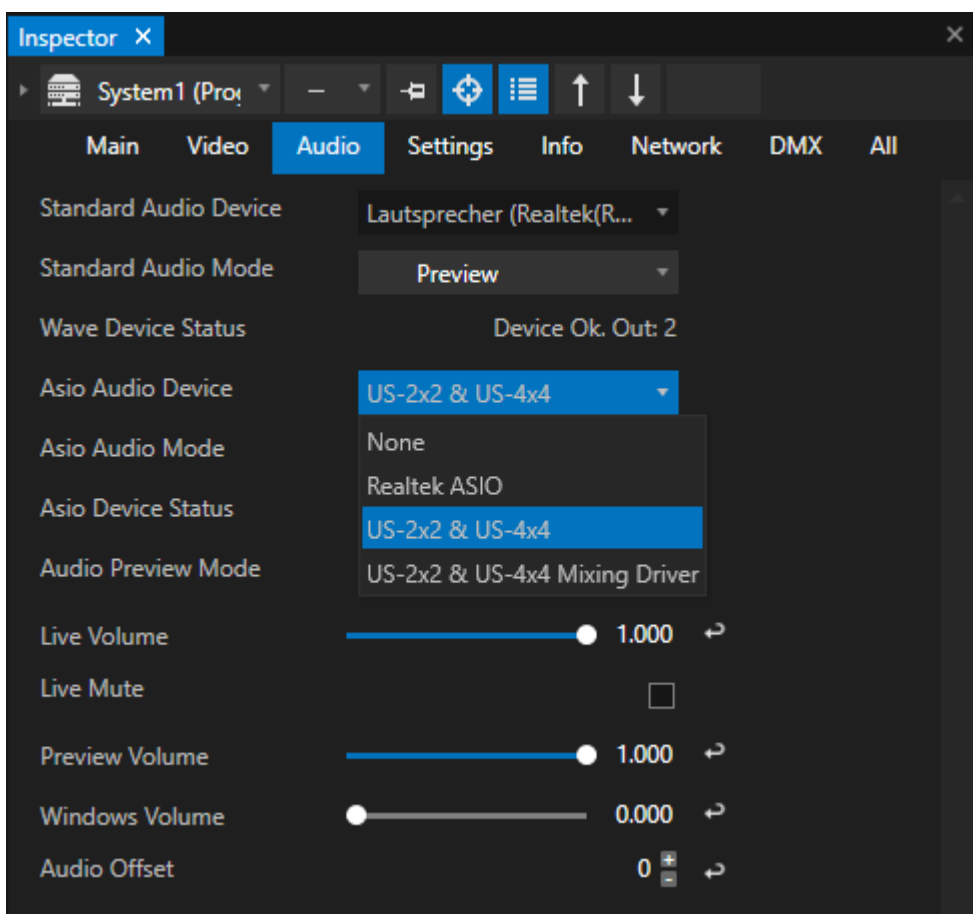
Recommended: plugged and connected Audio device/interface with ASIO driver  
VERTEX is clocked by the ASIO devices Clock.



### Audio and Video Playback with Audio over ASIO Interface

To prevent from a shift between audio and video we recommended to use the ASIO clock as system clock. The Video Playback and Frame processing in VERTEX is synchronized to the Clock signal from the ASIO device

When you select an ASIO device in the audio settings of a system, VERTEX automatically sets the Sync Clock from Wave Audio to ASIO Audio.



## Wave Audio

By default, VERTEX playback is locked to the clock of your Standard Audio device, which is in most cases your built-in sound card.

In the System Audio Settings, when changing the "Wave Audio Mode" from "Preview" to "Live", the Sync Clock automatically is changed to "Wave Audio"

## SMPTE IO LTC

Longitudinal Timecode can be received via ioversals [SMPTE IO Interface](#)  
Get all information about clock and playback settings in the [chapter SMPTE](#)

## GPU Sync

Syncs VERTEX Playback to the GPU Clock.

When using this option, you have to define a GPU sync master system. Go to Advanced Mode in the Inspector and search for "Is GPU Master" Property (Setting tab in the Inspector).

There is also an Advanced Property to set the "GPU sync interval" manually.

If there are questions about a GPU sync setup, feel free to [drop us an email](#) with your specific questions and a description of your hardware!

## Set Sync Clock In A Session Mode Project

Please read also the [Multi Systems](#) chapter. The topic [Playback Sync](#) explains how to set up the system clock in a Project.

All Systems in a session mode project, excluding the Clock Master System need to be set to the same clock. The Clock Master generates a System Clock for all other Systems. Please read the chapter [Playback Sync](#).

## 5.9 User Interface

- A new VERTEX project starts with a default workspace
- All windows and editors can be docked and/or undocked, reordered and repositioned
- All windows and editors can be opened multiple times - there is no restriction
- A user can store his window layout for a project as view

### The User Interface concept

When you have worked with any compositing software, media server or video editing software before, the Vertex interface should look familiar to you.

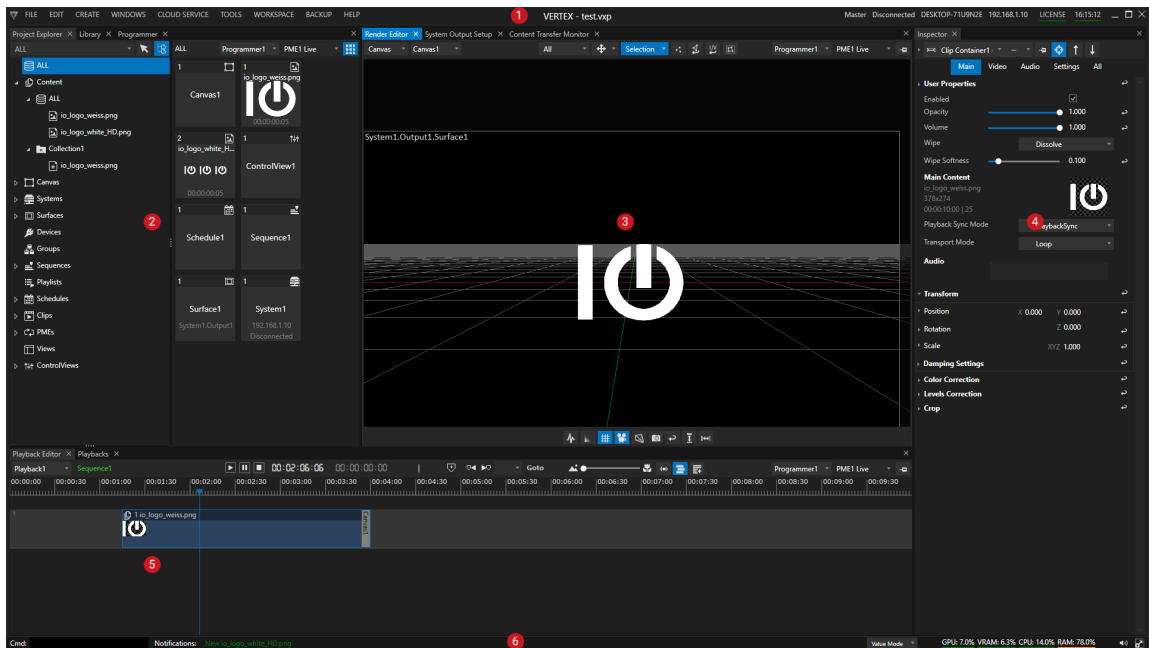
With VERTEX we want provide you a quick access to the main tools, so that you can proceed your daily work smart and fast:

- **functionality** that you **use less frequently** is located in **separate editor windows** that you could open when needed
- the workspace concept allows you to **build your own window arrangement** that matches to your project needs and routines.
- you are able to open and combine **more than one window and editor of the same type**. The data and settings are synchronized.
- **Context menus** give you a fast access to functionality
- The **settings** of an element are centrally located in **the Inspector**

### Default Workspace

Every new VERTEX project starts with the default workspace.

This default workspace is subdivided into the task-oriented sections that are described below:



<p><b>1</b></p>	<p><u>Top Bar</u></p>	<p><b>Main menu</b>  <i>file and project management, project settings, import content, create new project objects like a sequence, a canvas; access to libraries, windows and editors; access to tools and help</i></p> <p><b>Project Name</b></p> <p><b>System Status</b>  <i>IP, system name, session status, clock</i></p> <p><b>License Management</b>  <i>license status and license options</i></p>
<p><b>2</b></p>	<p><u>Content and Project Management</u></p>	<p><b>Project Explorer:</b>  <i>quick access to all local and network resources on the left, all project specific resources on the right</i></p> <p><b>Library</b>  <i>Access to video fx, devices and geometry modifiers</i></p>
<p><b>3</b></p>	<p><u>Visual Workspace</u></p>	<p><b>Render Editor</b>  <i>Your working area and preview section in 2D and 3D</i></p>

4	<u><a href="#">Settings and Information</a></u>	<p><b>Inspector:</b> The inspector bundles all settings, information and parameters. It is the central point to modify parameters and inform about project elements.</p>
5	<u><a href="#">Playback Editing and Management</a></u>	<p><b>Playback Editor:</b> Edit clip containers and cues, set keyframes, arrange clip containers on tracks.</p> <p><b>Playbacks Window:</b> Manage and overview all playbacks and their current play status.</p>
6	<u><a href="#">Status Bar</a></u>	<p>System health and monitoring, system notifications, quick access and manipulations with script commands, switch between value and programmer mode, set to fullscreen.</p>

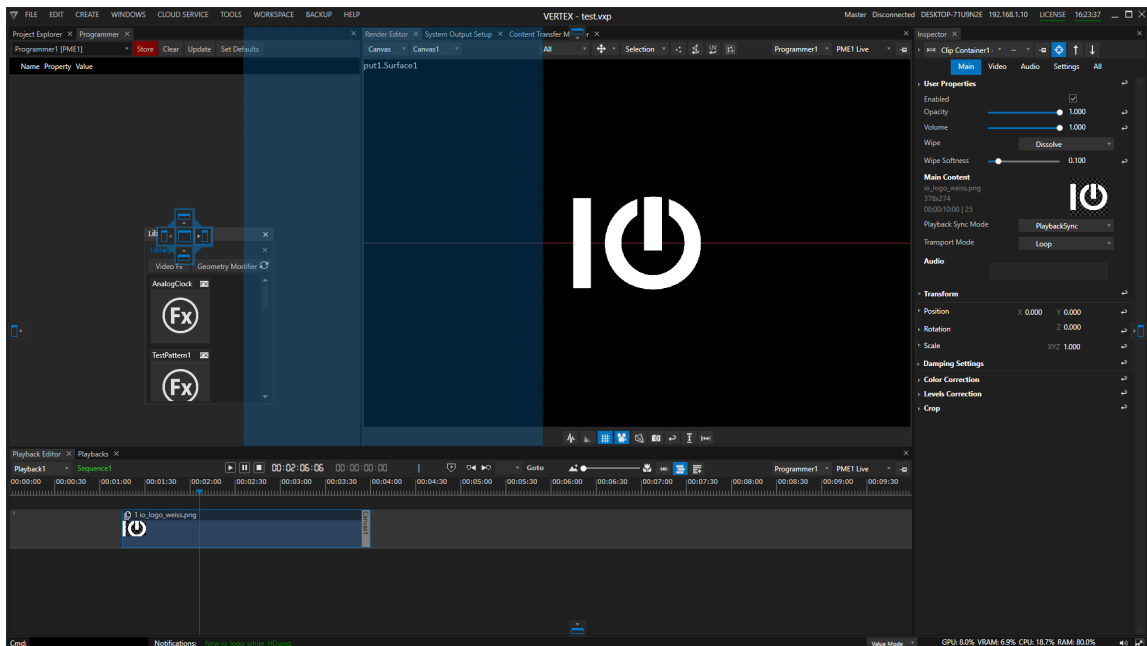
## Docking and Undocking of Windows and Editors

There **3 different ways** to personalize your workspace and to arrange and to dock windows and editors:

1. **As a tab** into the existing workspace layout
2. **At a new position** into the **applications main frame**
  - between some windows
  - at bottom sides or top
3. As **floating window** that is not docked in the main application frame.  
Such kind of floating windows you can freely move on your desktop.

Blue **helper widgets support you** with docking and suggest positions





When moving a floating window over the main application frame, a blue docking widget helps you and proposes docking positions.

Editors that are once arranged could be undocked again, re-arranged or deleted. Just click to the windows tab symbol and drag it out of its position.

## Multiple Editors of the same type

Your are able to open multiple windows/editors of the same type like Inspectors, Render Editors, Project Explorers.

## Views

Your customized workspace for a project can be saved as view into your project file.

### Clear a view

1. Go to the top bar
2. select "Workspace" in the main menu

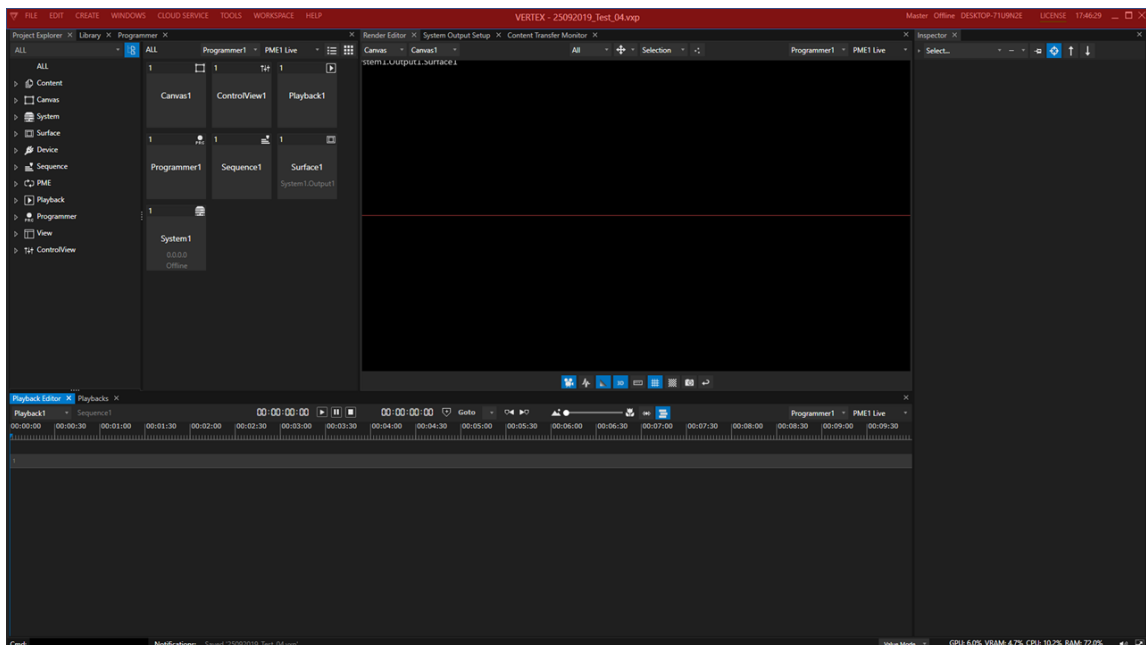
3. select "Clear View"
4. The workspace is cleared and empty - no editors and windows are displayed
5. Arrange a new view by open editors and windows with the "Windows" menu on the Top Bar

## Load/Reset to default workspace

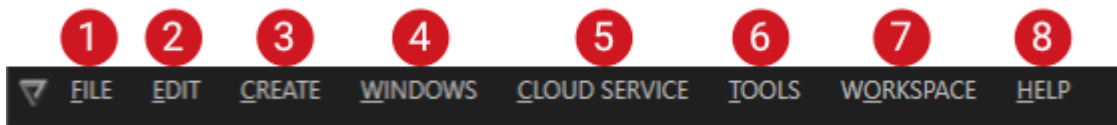
1. Go to the top bar
2. select "Workspace" in the main menu
3. select "Live"

### 5.9.1 Top-Bar

- The top bar gives you access to the **main menu** and the main **system information** - it is split into the following parts:
  - **main menu** on the left
  - current **project name** in the middle
  - **system information** on the right
  - **license management** on the right



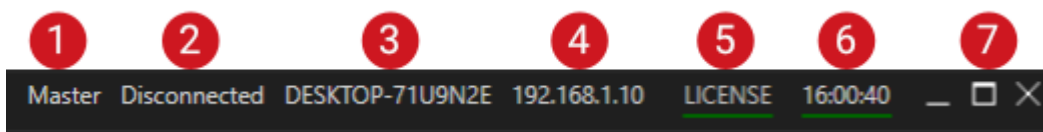
## Main Menu



<b>1</b>	<b>File</b>	<b>File Handling:</b> Save Project Load Project Import Files Connect Systems
<b>2</b>	<b>Edit</b>	<b>Edit options:</b> Projects Settings Copy/Paste Undo and Redo
<b>3</b>	<b>Create</b>	<b>Create Items</b> Import Content Import Image Sequences Create Generative Content Create all available Project Items like Surfaces, new Scripts, new Canvas ....
<b>4</b>	<b>Windows</b>	<b>Create new Windows and Editors</b> Open to create a new Editor or Status Window
<b>5</b>	<b>Cloud Service</b>	not available in current VERTEX Version - will come in future releases
<b>6</b>	<b>Tools</b>	not available in current VERTEX Version - will come in future releases
<b>7</b>	<b>Workspace</b>	Switch or Reset Workspace
<b>8</b>	<b>Help</b>	<b>Different Support and help options like</b> Create Ticket Show Serial

		<p>Open Task Manager</p> <p>Open Log Files ....</p>
--	--	---

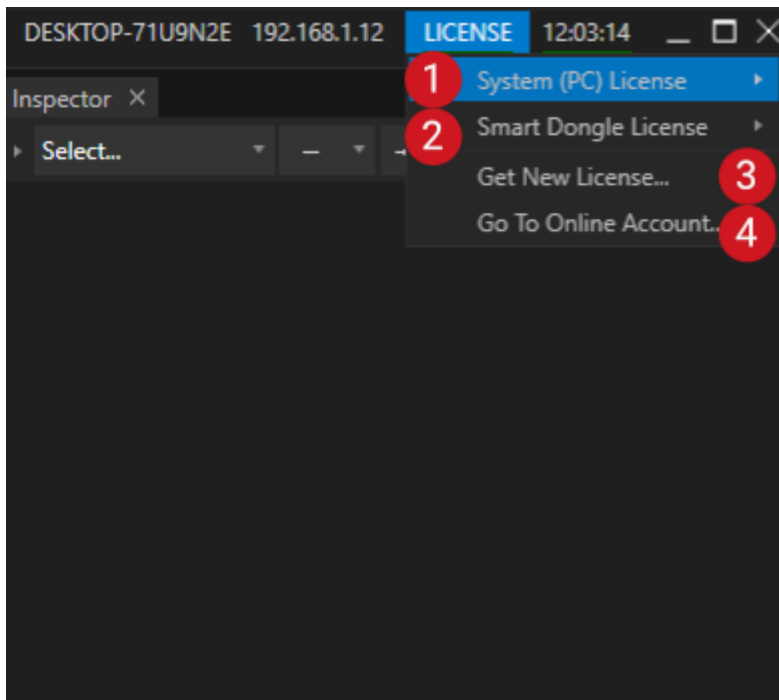
## System Information



<p><b>1</b></p>	<p><b>System role</b></p>	<p>Shows the <b>Project Role</b> of a System</p> <p>When only working with a single System: Always Master.</p> <p>When Working into a Session with multiple VERTEX Systems: Master or SessionMember</p>
<p><b>2</b></p>	<p><b>VERTEX Session Status</b></p>	<p>Shows the <b>status of your local System when working with multiple VERTEX Clients into a Session</b></p> <p>When working with only 1 System: Always Disconnected</p> <p>When working into a Session: Connected or Disconnected</p> <p><b>Note:</b> Please do not mix up with your internet or general Ethernet status. This status field only is related to the project online status for a Session with Multiple VERTEX Systems</p>
<p><b>3</b></p>	<p><b>Local Computer Name</b></p>	<p>Shows your Computer name that was set in Microsoft Windows System Settings for this hardware.</p>
<p><b>4</b></p>	<p><b>Local System IP Address</b></p>	<p><b>IP Address of your local VERTEX System.</b></p> <p>Shows the IP settings for the default network adapter that was set into VERTEX. If this field is empty or an IP Address of another, wrong adapter is shown, please check Settings for your VERTEX System into Inspector or check if default network adapter was set for your VERTEX System.</p>
<p><b>5</b></p>	<p><b>License Management</b></p>	<p>Mouse-click opens Menu for License Management.</p> <p><b>Status Colors:</b></p>

	<i>dropdown and license status</i>	<b>Green:</b> Valid and activated License <b>Red:</b> No License activated
<b>6</b>	<u>System Clock</u>	<b>Shows current System Time</b> <i>Or for a Session Member: Shows Project Time from Clock System</i>  <b>Status Colors</b> (only relevant if working into a VERTEX Session): <b>Green:</b> System in Sync - Sync Clock works <b>Red:</b> No Sync clock data received - please double check all Sync Clock Settings
<b>7</b>	VERTEX Window - Minimize, Close, reduce	Minimize, Maximize or Close VERTEX main Application Window

## License Management



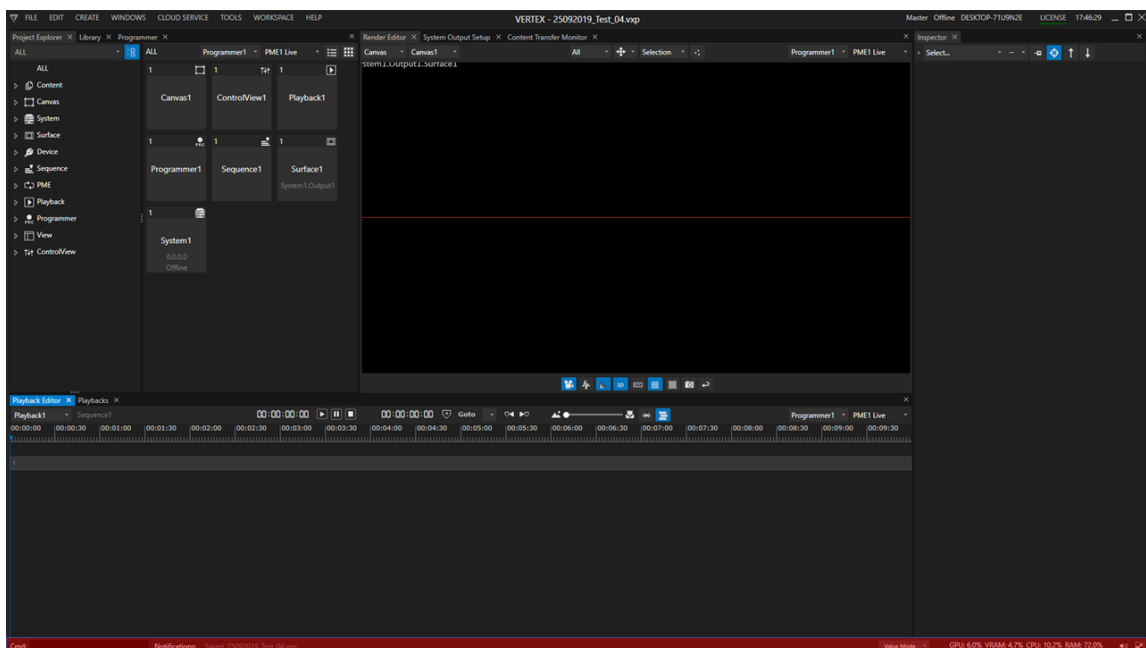
Activate and deactivate your VERTEX License or transfer it to a Dongle.  
Please look at [License Activation](#) to find a detailed description.

<p><b>1</b></p>	<p><b>Online License Activation</b></p>	<p>For the following 3 points an internet connection is required:</p> <ul style="list-style-type: none"> <li>License Activation</li> <li>License Update</li> <li>License Deactivation</li> </ul> <p>Information about your License State</p> <p>License Info</p>
<p><b>2</b></p>	<p><b>Dongle Activation</b></p>	<p>For the following 3 points an internet connection and a plugged in empty ioversal smart dongle is required:</p> <ul style="list-style-type: none"> <li>Dongle Activation</li> <li>Dongle Update</li> <li>Dongle Deactivation</li> </ul> <p>Requires a plugged in ioversal smart dongle</p> <p>Dongle Info</p>

3	<b>Offline License Activation</b>	No internet Connection but an external USB Stick/SD Card or Drive required for file exchange Export License Activation File Export Dongle Activation File
5	<b>ioversal Online Services</b>	Internet connection required

## 5.9.2 Status Bar

- The Status bar gives you a quick overview about **the current System hardware usage (RAM, CPU, GPU), global audio levels and timecode**
- Quick access to project **notifications** and their history
- **Command line section** to enter **Scripts**
- Switch to enable/disable the **Fullscreen Renderer** and to **mute/activate audio preview**
- Switch between **Value Mode** and **Programmer Mode**



## User Interface



1	<b>Command Line Section</b>	quick access to the <a href="#">script</a> console.
2	<b>Notifications</b>	Notification and information from VERTEX <b>Actions:</b> left-click on the notification section to open an overlay window with a chronological history of the last System notifications
3	<b><a href="#">Inspector Mode</a></b>	Switch to Advanced Inspector Mode to show more properties of an item in the Inspector Most of the properties in Advanced mode are for edge-cases. To keep the Inspector more clean for your daily use-cases, VERTEX offers this filter option <b>Default:</b> Standard
4	<b><a href="#">Value or Programmer Mode</a></b>	Switch between Value Mode (Default) and Programmer Mode and choose how you prefer to handle and manipulate values
5	<b>Hardware usage and Performance</b>	System hardware usage: displays the current usage of your GPU (total sum and VRAM usage), your CPU (total sum of all cores), and your System Memory (RAM). <b>If the Fullscreen Renderer is enabled</b> on one of your screens, current render performance is displayed in FPS
6	<b>Mute Preview Audio</b>	Mute or enable your local Systems <a href="#">preview audio</a> output Default: on
7	<b>Enter Control Viewer in Full Screen</b>	Starts the local <a href="#">Control Viewer</a> in Full Screen quit with CTRL + F5
8	<b>Enter Full Screen Renderer</b>	starts the <a href="#">Fullscreen Renderer</a> on your local System quit Fullscreen Renderer with CTRL + F or ESC This button is not available in <a href="#">VERTEX Touch Edition</a>



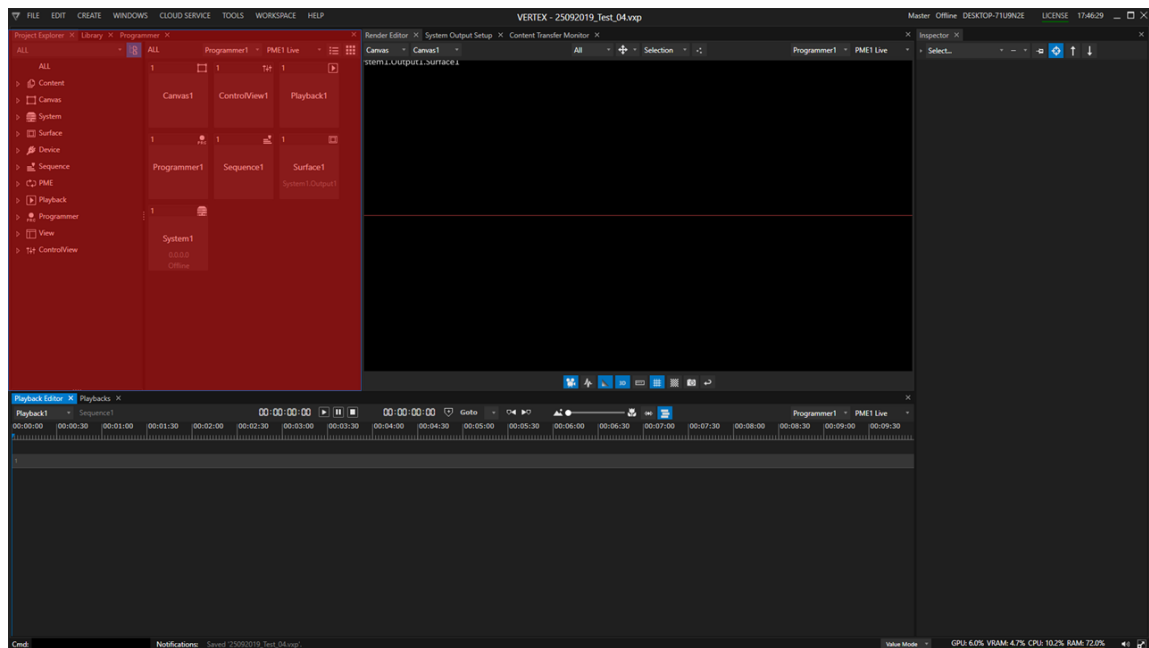
**GPU Usage in status bar will not work with AMD and Intel Graphic Cards**

When your PC has an AMD GPU inside, the GPU usage information in the status bar is not available.

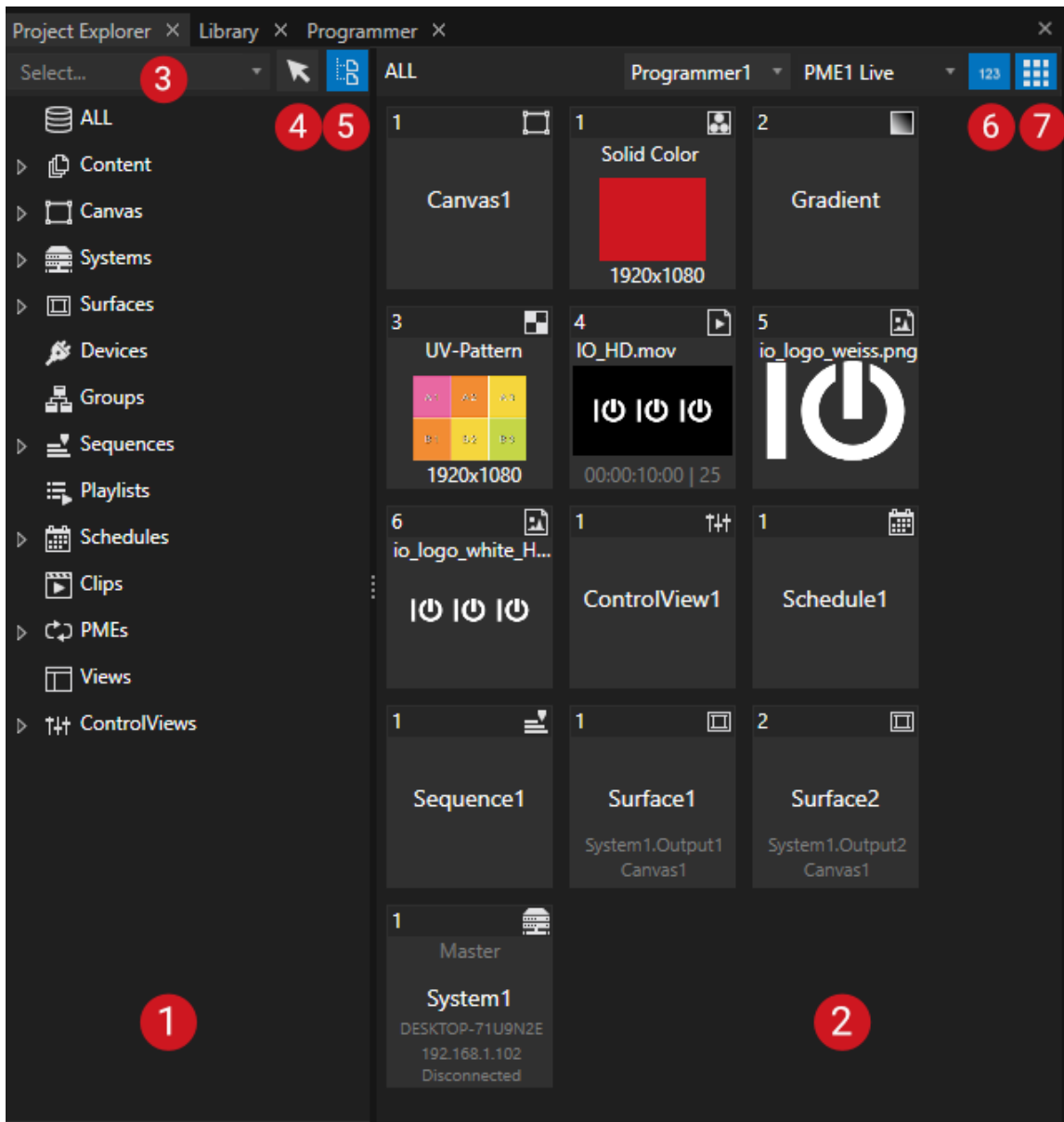
When using an AMD or Intel GPU, VERTEX will only show CPU and RAM usage.

### 5.9.3 Project Explorer

- The Project Explorer gives you the **overview about project resources**
- The **left pane** shows you all items and elements of your project - sorted by object category and its child elements
- The **right pane** shows you all child elements of a selected object category - e.g. all Content Assets
- **Context Menus** give you access to item specific functionality



## User Interface



The Project Explorer: Overview about all available elements and resources on the left split and about all elements of your current project on the right split

<p><b>1</b></p>	<p><b>Project Tree</b></p>	<p>The left split of the Project Explorer shows you <b>all the available resources and elements from your current System</b> on which Vertex is running and <b>all items that are part of your project</b></p> <p><b>Actions</b></p> <ul style="list-style-type: none"> <li>• Selection with left mouse buttons to show project related child elements in right project column</li> <li>• Right mouse click to open the context menu with additional options - the options differ for each element type</li> <li>• Drag an element from the tree view to the right pane or a Collection</li> </ul>
-----------------	----------------------------	--

		<ul style="list-style-type: none"> <li>• Drag with your mouse to favorite folder or parent folder in tree view</li> <li>• Select an element to focus on it in the Inspector</li> </ul> <p><b>Style options</b></p> <ul style="list-style-type: none"> <li>• Filter by element type <b>3</b></li> <li>• show/hide child elements <b>5</b></li> </ul> <p><b>Selection options</b></p> <ul style="list-style-type: none"> <li>• Focus Tree Items into Inspector. Default: off <b>5</b> When switched on, items in the Project Tree that are selected with your mouse, will be shown into Inspector</li> </ul>
<b>2</b>	<b>Project items</b>	<p>Shows all elements and assets which are part of your project Organizes Assets, Items and Objects by Categories and Child Elements</p> <p><b>Actions:</b></p> <ul style="list-style-type: none"> <li>• Select one item to set property values in Inspector</li> <li>• Select multiple items to collectively set property values in the Inspector (e.g. Opacity for all selected Content Items to 0)</li> <li>• Right-click with your mouse to open a context menu with additional options</li> <li>• drag and drop elements</li> </ul> <p><b>Sorting</b> <b>6</b></p> <ul style="list-style-type: none"> <li>• By Content Item ID (default)</li> <li>• Alphabetically descending</li> </ul> <p><b>Style options:</b> <b>7</b></p> <ul style="list-style-type: none"> <li>• Tile view (default)</li> <li>• List view</li> </ul>



#### **Drag and drop elements**

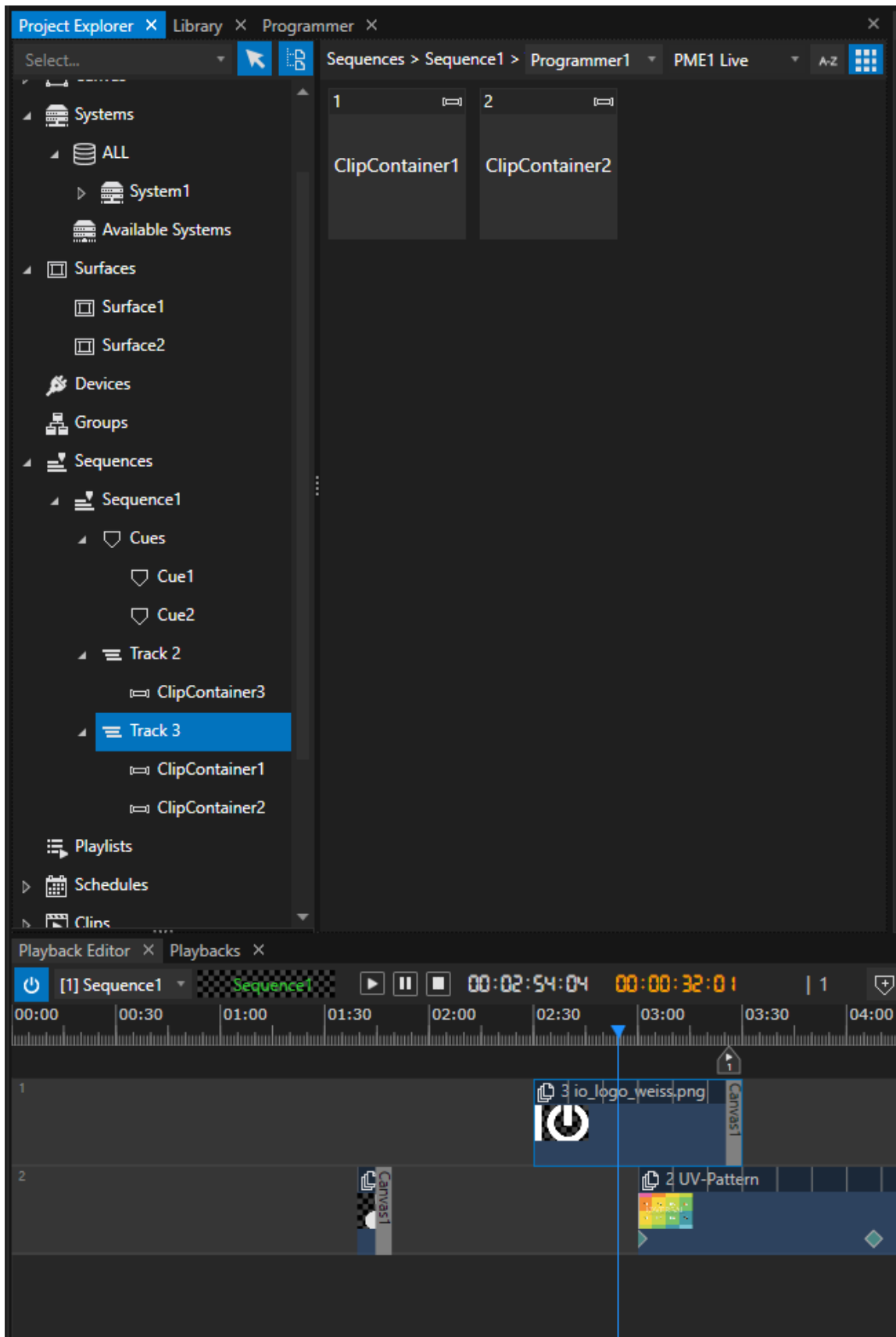
Use **drag and drop instead of context menus**. Just drag and drop e.g. an NDI receiver of System1 from the left element tree to the right pane to include it into your current project. For those elements where a logical context makes no sense (e.g. drag a System and drop it on another System) you will get visual feedback on the target element next to your mouse pointer

**Drag and drop does not only work in the Project Explorer.** You are able to drag content from the Project Explorer and drop it into the Render Editor or the playback editor.

You are able to drag content from Windows file explorer to your VERTEX Project Explorer

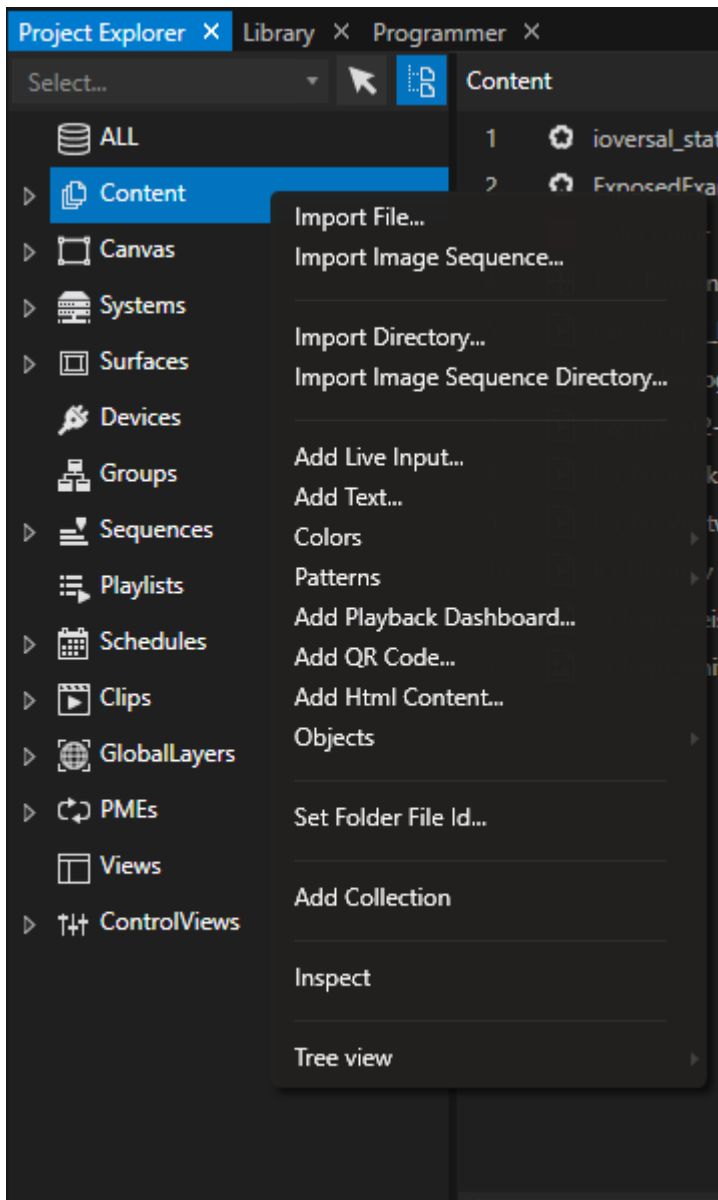
## Access to Child Elements

- The Project Explorer is organized in main Object Categories and its children
- With this structure, you are able to access e.g. a Track of a Sequence to do some specific setting
- An arrow indicates whether you can still dive deeper into the child structure



## Context Menu(s)

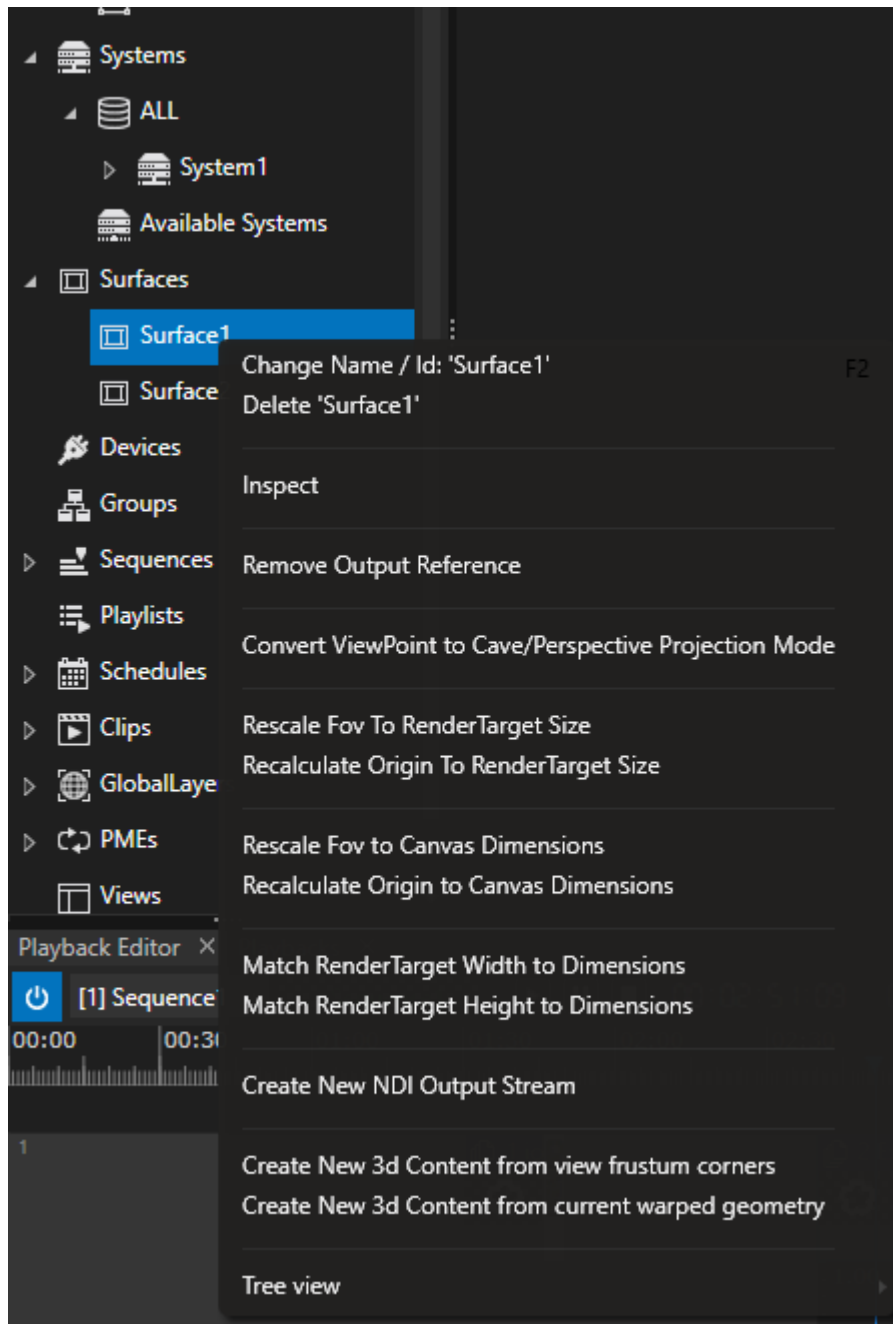
- Use Context Menus into the Project Explorer to have access to some item **specific functionality**, to **quick access settings** or just to **add a collection** or to **add a new element**
- Context Menus **differ depending on the item type**
- **Some functions can only be accessed via context menus in Project Explorer -e.g.Re-Scan Video Inputs for a System** (or of course with a Script)



Open the Context menu for Content with a right-click on the Content category.  
Add or import Content

## Examples

- Select an available System into your network ( in the left project tree) and use the context menu to remotely start VERTEX on this System
- Add 4 Surfaces as horizontal/vertical split as child of an output
- Refresh video sources



Context menu for a Surface into a Project: very specific functionality and quick access to e.g. Rescale FoV

## Shortcuts

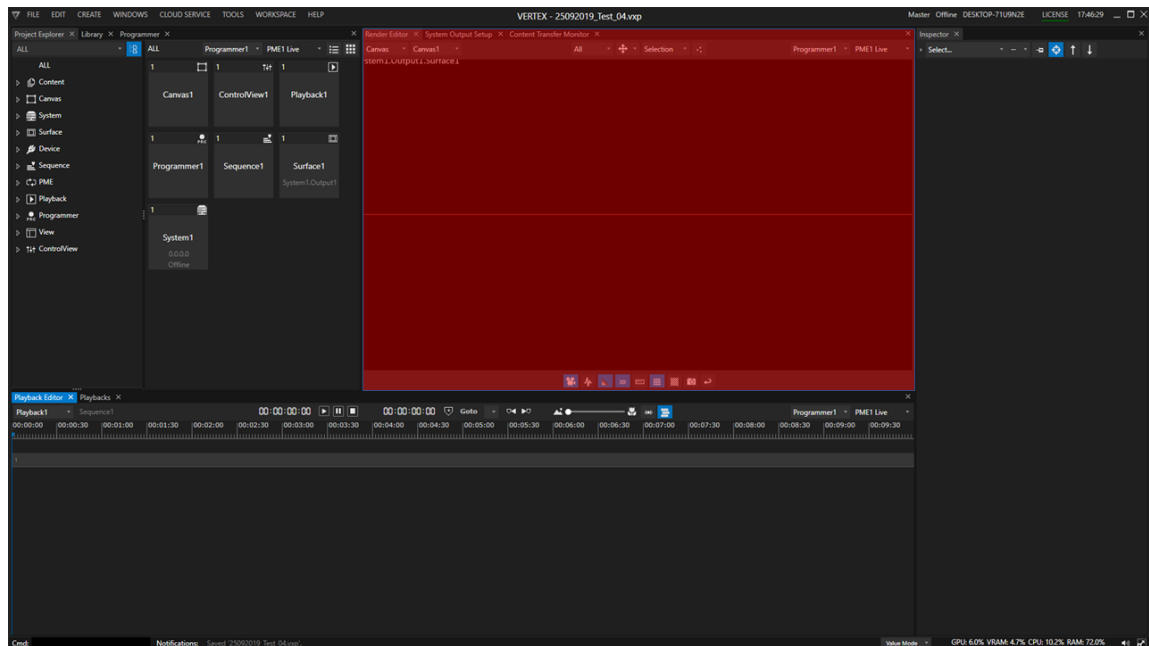
Shortcut	Action	Notes
CTRL + I	Import Content	



F2	Rename Element	

## 5.9.4 Render Editor

- The Render Editor is the **visual preview**: arrange, select and manipulate the items of your show.
- Work with **different views** and switch between them - a view that is **focused onto your Canvas** or another one that is **focused on your System outputs**
- Open **multiple Render Editors and work with them in parallel** - the performance limit depends on your hardware



## Navigation

There are different options to navigate through the 3D/2D space of VERTEX's Render Editor. Depending on your personal preferences, you can use

### **1. Left Mouse Button and Keys 1,2,3**

Left Mouse Button + 1: Pan/Move left/right/up/down

Left Mouse Button + 2 : Zoom/Dolly in and out

Left Mouse Button + 3 : tilt

### **2. Middle Mouse Button/Wheel ( and Alt-Key)**

Middle Mouse Button/Wheel pressed: Pan/Move left/right/up/down

Mouse Wheel: Zoom/Dolly in and out

Alt+ Middle Mouse Button/Wheel pressed tilt

OR

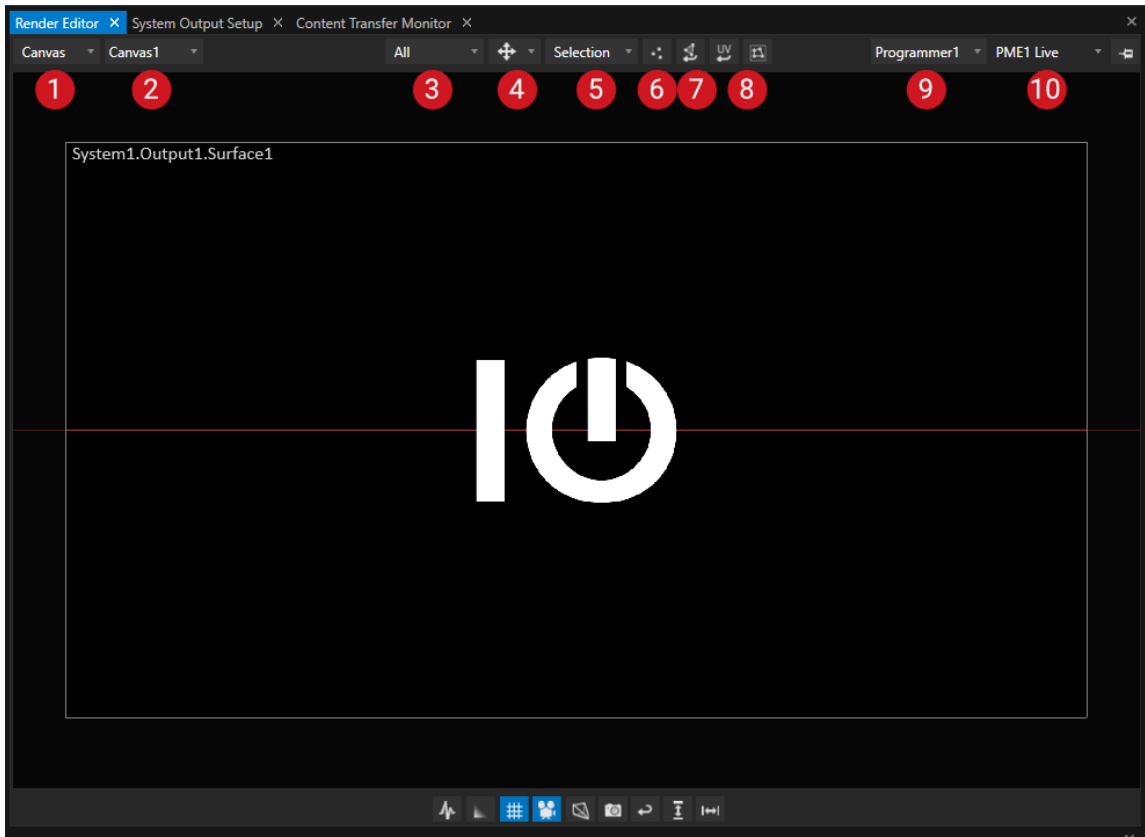
### **3. WASD - Keys (or Arrow Keys)**

W or S Key: Zoom/Dolly in and out

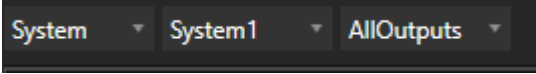





A or D Key: Pan/Move left/right

Q or E Key: Tilt left/right

## User Interface



<p><b>1</b></p>	<p><b>View Mode</b></p>	<p><b>Default:</b> Canvas view  <b>Options:</b> All Canvases, Canvas, System, DMX Mapping, Surface Map, Visualizer</p> <p><b>Select your workspace view:</b>          Canvas View lets you select a canvas to work with. Arrange and compose your content and build your scene from here.          System View shows the final image of your output(s).          Good for controlling the final signal of each output or to make some visual modifications on output/Surface level</p> <p>refine your selection with the "select specific view" dropdown <b>2</b></p>
<p><b>2</b></p>	<p><b>Select specific view</b></p>	<p><b>For View Mode Canvas:</b>          When you are working with more than 1 Canvas into your project, you can switch between all Canvases of your project</p> <p><b>For View Mode Output</b>          when select this view mode another selection box is displayed</p>

		 <p>System: When working with only one System dont care about this. When working with more than one Systems into your project, you can select the System whose Output you want to modify/show</p> <p>Output Selection: All Outputs - general overview with the content of all outputs of one selected System Output # - shows you one specific output for preview or further modifying (also see. <a href="#">Canvas, Surface, Output</a>)</p>
3	<b>Element Selection Filter</b>	<p>Options: All(default), Surface and Content</p> <p>Filter if you want to select and show all elements into a Canvas or select only surfaces or content.</p> <p>Useful when you don't want to accidentally touch some surfaces that already are in final position</p>
4	<b>Mouse Mode</b>	<p>Select the functionality/mode your mouse courser should have when working into the Render Editor</p> <ul style="list-style-type: none"> <li> Only Select Content</li> <li> Move selected Content (Default)</li> <li> Rotate selected Content</li> <li> Scale selected Content</li> <li> Touch interaction for interactive elements</li> </ul>
5	<b>Select whole element, vertices or Modifiers</b>	<p><b>Options:</b></p> <p>Select Whole element</p> <p>select vertices of an element</p> <p>Select Modifiers ( c.f. <a href="#">Geometry Modifiers</a>)</p>
6	<b>Select covered vertices</b>	<p><b>Default:</b> off</p> <p>When switched to on, also hidden surfaces of a 3d model in the depth axis will be selected</p>
7	<b>Reset VERTEX Offsets</b>	<p>Resets all modified Vertice offset to default value</p>

8	<b>UV Modifiers</b>	Resets all UV Offsets to default value or Apply UV modification from current View as setting
9	<b>Programmer selection</b>	When working in programmer mode: Select programmer where your changes should be assign to
10	<b>PME (Playback Mixing Engine) Selection</b>	Select PME (Playback Mixing Engine) you want to display in the Render Editor

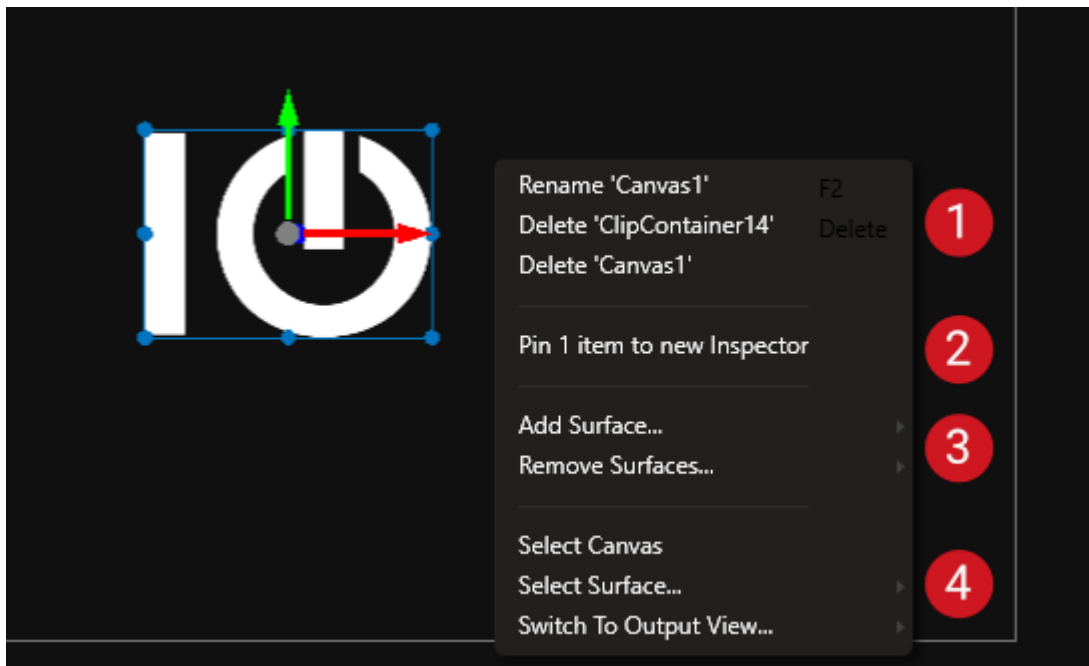
Bottom bar



1	<b>Show Statistics</b>	<p><b>Default:</b> off <b>Option:</b> enabled</p> <p>An overlay is displayed in the left upper corner of the Render Editor that shows you</p> <ul style="list-style-type: none"> <li>• the current render performance of this specific Render Editor in Frames per Second (FPS)</li> <li>• The global sync clocks uptime</li> <li>• a flashing white square for optical sync test</li> </ul> <p>This overlaid information is not visible in the Fullscreen Renderer and can be set for every Render Editor separately</p>
2	<b>Switch between Full Resolution or Proxy Mode with lower resolution</b>	<p><b>Default:</b> on - Full Resolution Mode is enabled. The content is rendered in its full resolution into the Render Editor</p> <p><b>Option:</b>- off - Proxy Mode with lower resolution The content is shown as Proxy Files with lower resolution. This gives you the options to work and preview even bigger shows with a lot of content in it</p>

		<p>or to pre-program with lower hardware without the original source files on it.</p> <p><b>Keep in mind:</b> When you has disabled the automated proxy encoding into the <a href="#">Project Settings</a>, no proxy files are available</p>
3	Show Grid	<p><b>Default:</b> on</p> <p><b>Option:</b> off - disables base grid into Render Editor</p>
4	Toggle between free and local camera	<p><b>Default:</b> on - local - the camera of your Render Editors workspace works locally in this window</p> <p><b>Option:</b> disabled - free: - each Render Editor where this option is disabled is synchronized in workspace mouse movements - also on different Systems into a network</p>
5	Show Surface Frustum	<p><b>Default:</b> off</p> <p><b>Option:</b> on - shows frustum and viewpoint origin for all Surfaces into Canvas</p>
6	save snapshot	<p>Save the current view into your Render Editor as png file and use this to send previews out to clients or add them into a storyboard on click: An explorer window opens, you can select the path and filename.</p>
7	reset view	<p>resets the current workspace view to the default values</p>
8	Vertical fit local view	<p>Adapts local camera view that it fits to Canvas height</p>
9	Horizontal fit local view	<p>Adapts local camera view that it fits to Canvas width</p>

## Context Menu

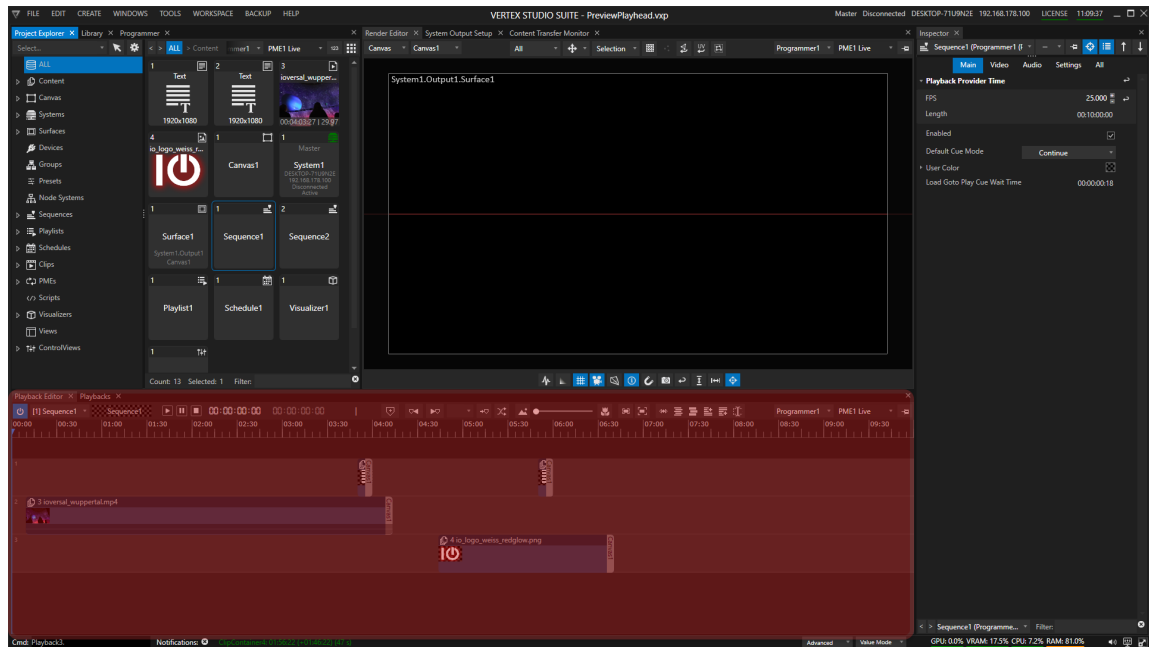


1	<i>Context related options</i>	<i>Depending on your selection - e.g. Delete object or rename object</i>
2	<i>Pin selection to new Inspector</i>	<i>Pins the selected item to a new Inspector window</i>
3	<i>Add or Remove Surface(s)</i>	<i>Add or remove surfaces to your Canvas. Add: shows all surfaces of your project which have not yet been added to your Canvas Remove: shows all surfaces which have already been added to your current Canvas</i>
4	<i>Quick Selection</i>	<i>Quick selection options Select Canvas: selects the Canvas of your current Render Editor view in the Inspector Select Surface: selects a surface in Render Editor and in the Inspector Switch to output view: fast switch to show an output without using the Render Editors dropdown lists Change view in Render Editor to Systems- selected output.</i>

## 5.9.5 Playback Editor

- The Playback Editor is your **main editing** window for building a **timeline and arranging your show**
- All content (audio, video, devices) in a sequence or playlist is arranged in **Clip Containers**.

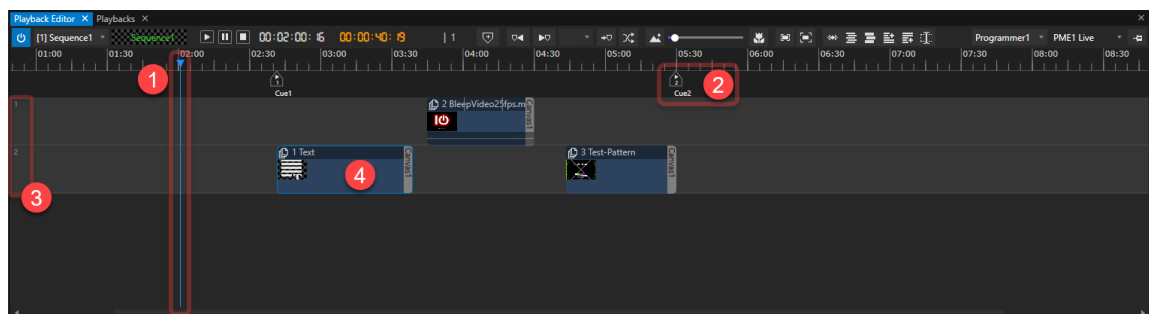
- *Clip Containers* can be moved around freely, as they are not time-locked like in common video editing suites.
- *Use individual tracks* to arrange and layer your *Clip Containers*.



## User Interface

- *The Playback Editors UI is timeline based.*
- *Arrange and layer your Clip Containers using separate tracks*
- *Cues are markers that have various functions in show control*

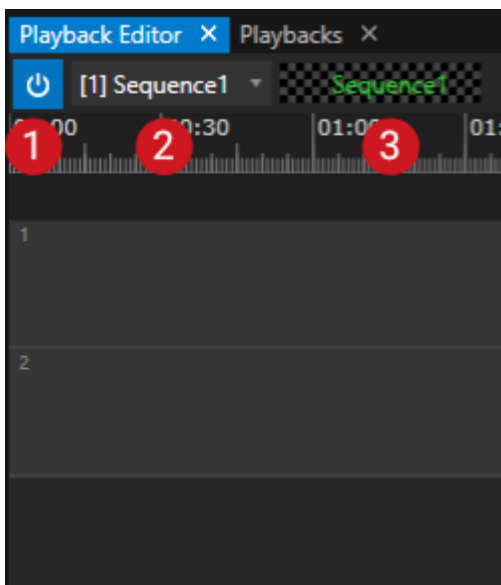
## Basics





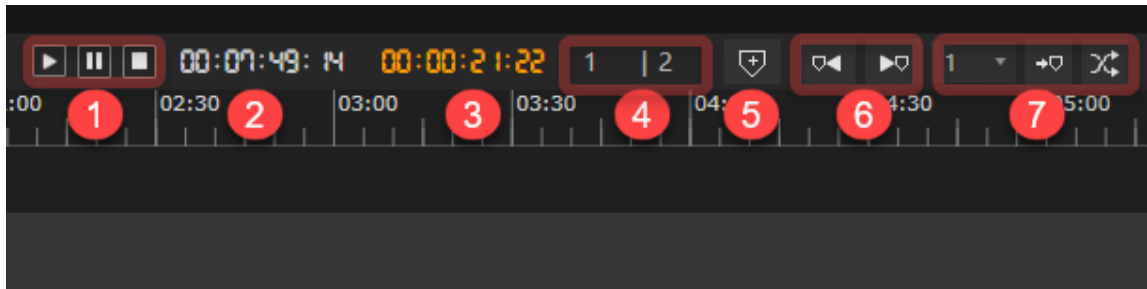
<b>1</b>	<i>Playhead</i>	<i>shows current Playback time/position</i>
<b>2</b>	<u><i>Cue</i></u>	<i>marker/step on the timeline</i>
<b>3</b>	<u><i>Track</i></u>	<i>arranging and sorting the layering hierarchy of Clip Containers in a sequence - top track is the first layer, always visible</i>
<b>4</b>	<u><i>Clip Container</i></u>	<i>host for different types of content, settings or devices</i>

## Playback and Playback Provider Selection

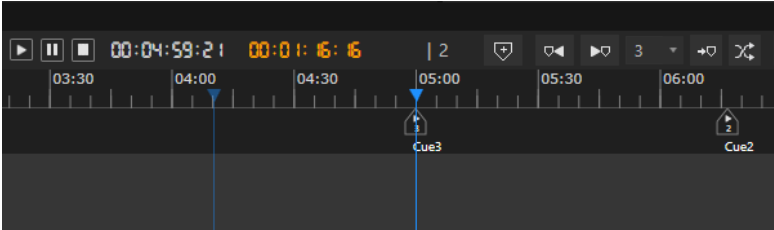


<b>1</b>	<i>Activate Playback</i>	<i>Default: on Switch off to deactivate a playback. When deactivated, no content is rendered.</i>
<b>2</b>	<u><i>Playback Selection</i></u>	<i>Drop-down menu allows the selection of separate playbacks.</i>
<b>3</b>	<u><i>Playback Provider</i></u>	<i>Shows the playback provider of this playback. Drag a sequence or playlist from Project Explorer and drop them into this field.</i>

## Playback and Cue Control



1	<b>Transport Buttons:</b> <i>Play, Pause, Stop</i>	<i>Starts, pauses and stops the playback.</i> <i>Stop button: playhead jumps back to the beginning of the sequence.</i>
2	<b>Current Playback Time</b>	<i>Shows the current timecode of the playback / the playhead's current position.</i> <i>Double-click to enter a time and the playhead will jump there.</i>
3	<b>Count-Down To Next Cue</b>	<i>Shows the remaining time until the next, upcoming cue.</i>
4	<b>Cue Number Status:</b> <i>Previous   Next</i>	<i>Shows the playback's position by cue numbers: Cue number before the playhead   and upcoming cue number.</i>
5	<b>Add Cue</b>	<i>Click to create a new cue at current playhead position.</i> <i>Continuous numbering of cues starts for each Playback with No 1. Number increases with every additional cue created.</i>
6	<b>Go to Previous/ Next Cue</b>	<i>Playback jumps back to previous / or forward to next upcoming cue on the timeline.</i>
7	<b>Select a Cue</b> <b>Jump to Cue</b> <b>Fade to Cue</b>	<i>Drop-down selection of a cue to jump or fade to. Alternatively, just enter a cue number and confirm with ENTER to jump.</i> <i>Makes the playhead <b>jump</b> to the cue selected in the drop-down menu.</i> <i>Renderer instantly shows content at cue position.</i> <i>Makes the playhead <b>fade</b> to the cue selected in the drop-down menu.</i> <i>Renderer crossfades to content at cue position.</i>






When using the `FadeToCue` command, the playhead will jump to the new point on the timeline, while its "ghost" will remain at the old position for the duration of the crossfade.

The duration of the crossfade can be set in the sequence's settings in advanced mode inspector under `Fade To Time`.

`FadeToCue` and `FadeToTime` are essentially the same functions.

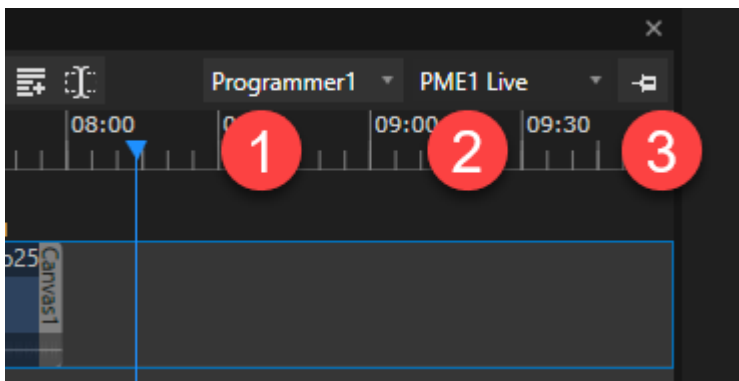
### Layout Options and Quick Access



1	<b>Zoom</b>	Zooms in or out of the timeline at position of the playhead.
2	<b>Zoom Reset</b> <b>Zoom To Item</b>	Zooms out completely. Zooms in on selected item(s).
3	<b>Auto Scroll</b>	 off: default  follow: sequence window refreshes when playhead runs past the displayed portion of the timeline.  center: the running playhead continuously stays in the center of the sequence window.
4	<b>Track Height</b>	click once to minimize/ maximize height of selected track(s), and double-click to adjust all tracks.

5	Add <u>Track on Top / at Bottom</u>	Adds a new track above or below.
6	<u>Cut all Clip Containers</u>	Edits (cuts) all Clip Containers at current playhead position.

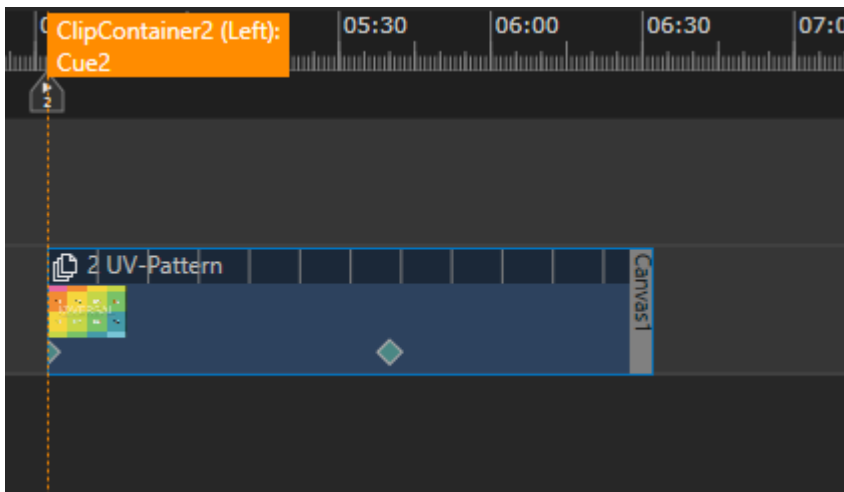
## Programmer and Playback Mixing Engine



1	<u>Programmer Selection</u>	When working in Programmer Mode, select the programmer to work with ( e.g. Live or Preview)
2	<u>Switch Playback Mixing Engine (PME)</u> between Live und Preview(s)	Switch whether your Playback editor should show the live environment or if you want to work in a Preview <b>Colors:</b> Grey (Default) - Live state Orange: Preview State
3	Track Live PME	When enabled and swapping live and preview PMEs, the editor window always follows the live PME.

## Snap

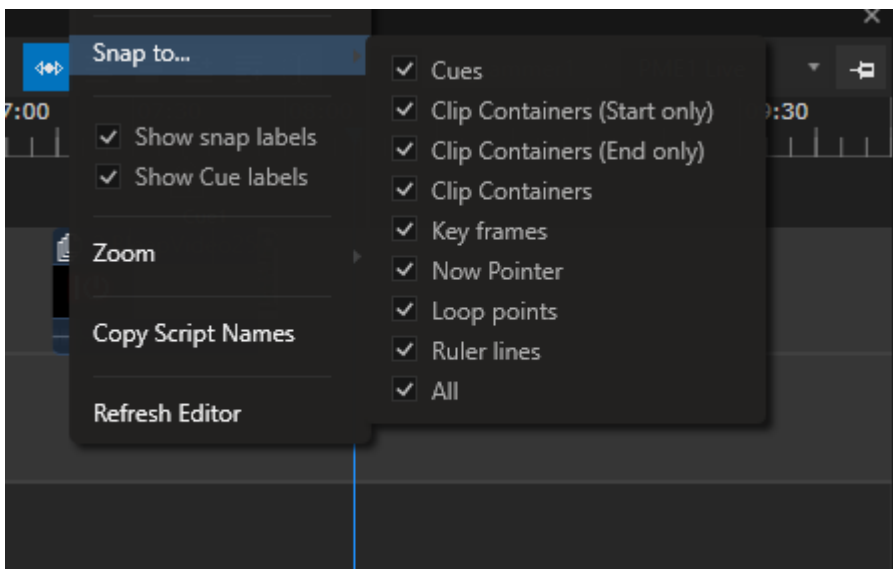
- The Playback Editor comes with an **smart snap function** that will help you with the **correct positioning** of Clip Containers and cues
- **Orange labels** show you the details of what you're currently snapping to.



Clip Container2 snaps to Cue2

## Settings

- By default the snapping is enabled to all elements
- You can customize it to your needs



- Right-Click with your mouse at any position into the Playback Editor and open the Context Menu
- Select "Snap to" and customize

## Working with the Playback Editor

- all content and all devices are arranged in **clip containers**
- **clip containers** can be plotted and layered **on tracks**, as well as **moved freely** between tracks
- a **Canvas or output** can be assigned as **render target** to each **clip container** - independent from the clip containers order on a track
- the **top track** is **rendered on top** (when the default settings of a clip container are used)

## Live and Preview Mode



As with the render editor, any playback editor has got two possible modes: **live and preview**.

The default mode is **live mode**: a blue colored playhead shows the playback's position at the live output.

The playback editor in **preview mode**

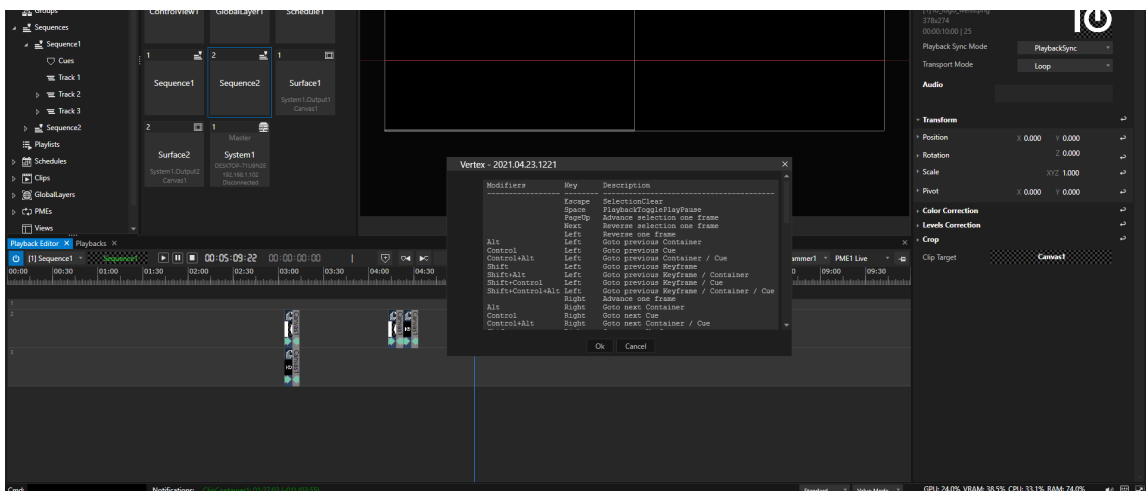
- is highlighted with **an orange border**

- has got an **orange colored playhead**
- shows the live mode playhead's position in a pale shade of blue.

[Click here for further reading on Playback Mixing Engine \(PME\).](#)

## Keyboard Shortcuts

- There are **seperate lists of all available shortcuts** for each editor and window in VERTEX.
- To keep the documentation up to date, the **shortcut lists are automatically created by the software itself**.
- Just **press the F1 Key** to open a list of shortcuts corresponding to the current window/ editor in focus.



Focus Playback Editor and Press F1 Key.  
A window with all available shortcuts for your Sequence opens.

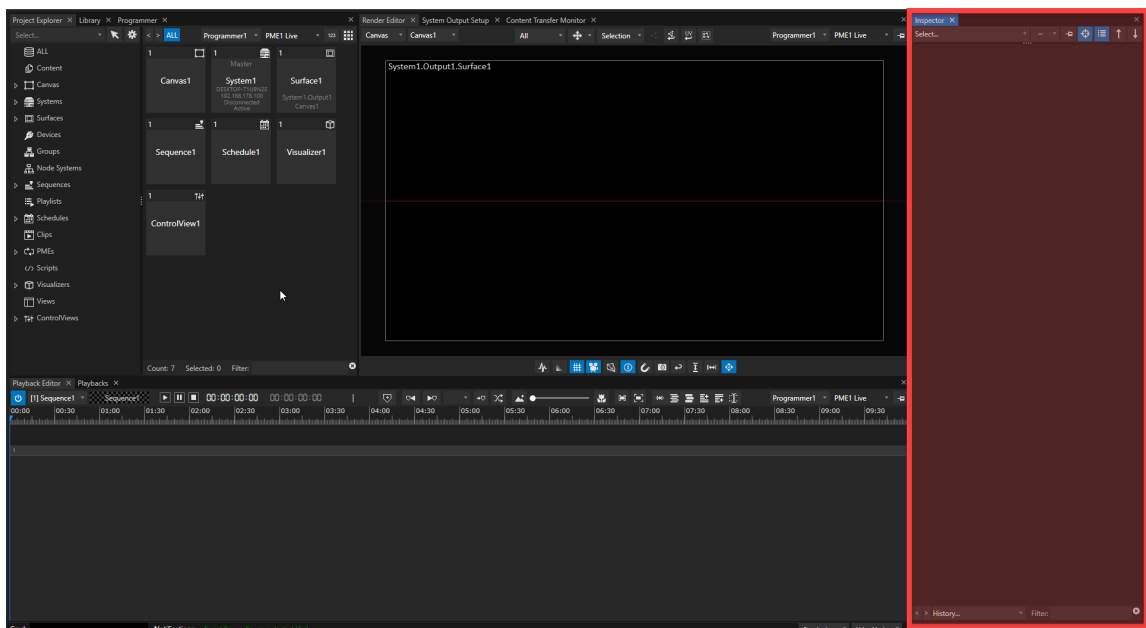
## Examples of Common Workflows:

- drag and drop content or devices on a track and automatically create a corresponding [Clip Container](#).
- [Video FX](#) or [Geometry Modifiers](#) can only be dropped on top of a Clip Container.
- select a Clip Container and edit all properties in the [Inspector](#)

- Pin a certain clip container to a new [Inspector](#) to edit properties
- add a track (go to context menu by right-click)
- Move Clip Containers on a track or between tracks (grab Clip Container with your mouse and move it around)
- snap Clip Containers to the playhead
- push or pull with the mouse on the edge of a Clip Container to change its length
- draw a multi selection rectangle with your mouse to select multiple containers to edit
- Copy/ paste Clip Containers (context menu or shortcuts)
- Double-click on a clip container to [edit Keyframes](#)
- insert a specific time between clip containers (go to context menu by right-click)

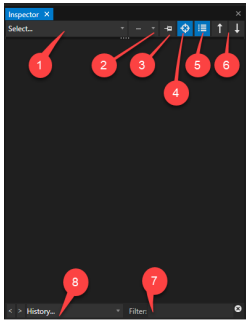
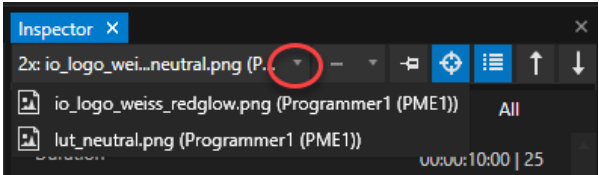
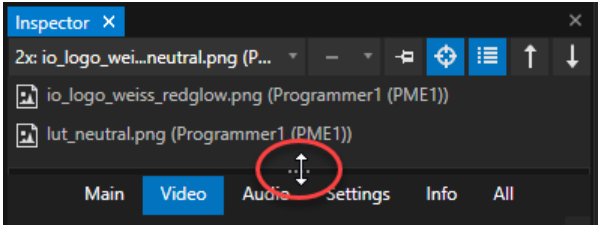
## 5.9.6 Inspector

- The Inspector window is the **central hub for all settings and property values in VERTEX**.
- **Click on any element** in the project explorer, render editor, sequence editor etc. **in order to inspect the element's properties and change its settings**.
- You can **work on multiple inspector windows simultaneously** and **pin items to an inspector**.
- **Tabs** help you to **sort settings**.
- The **viewing modes standard** and **advanced** filter out distinct properties and keep the view on the inspector neat and clean.





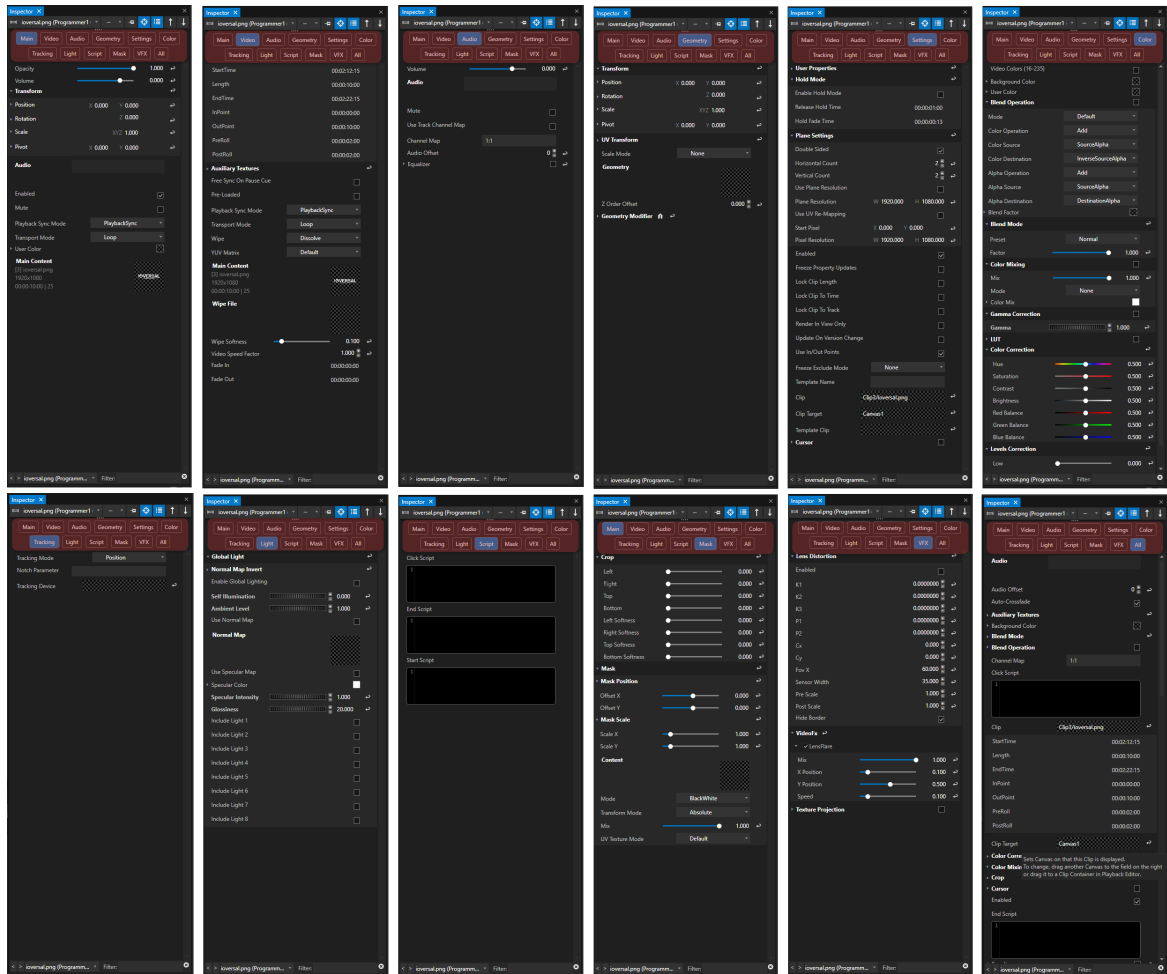
## User Interface

	<p><b>1</b></p>	<p>Selection</p>	<p>Displays the selected item's name. This selection is active in the inspector window. <b>View a list of your multiple selections</b> by either opening the drop down menu here,</p>  <p>or by pulling down the tab at dotted line:</p> 
	<p><b>2</b></p>	<p>Align</p>	<p>Enables different modes to change settings of multiple selected items in relation to one another. <a href="#">See below.</a></p>
	<p><b>3</b></p>	<p>Pin To Inspector</p>	<p>VERTEX's pin function has three modes:</p> <ol style="list-style-type: none"> <li>1. None is pinned (default) - meaning whenever you select a new item, it appears in the inspector.</li> <li>2. Items are pinned - meaning the current inspected item is pinned to this inspector; a different item selected for edits will not show up in the inspector.</li> <li>3. Context is pinned - meaning only items from the current window/ editor will show up in the inspector.</li> </ol>
	<p><b>4</b></p>	<p>Selection Focus</p>	<p>Toggles between two selection modes: the target icon enables selections just from the current editing window focus. The globe icon enables across all the different editing windows in VERTEX - for instance, in this mode you can select multiple items from the project explorer, playback editor and render editor simultaneously.</p>
	<p><b>5</b></p>	<p>Multiple Selection</p>	<p>The list icon enables multiple selections. The single item icon disables it.</p>

	6	Prev/ Next Arrows	Jump back and forth between multiple selected items in the list view. See instructions above, list view <b>1</b> .
	7	Filter / Search Function	The filter is a helpful searching tool when working with items that have lots of properties. Gain quick access to the property you need by <b>typing in a letter or a word</b> and <b>confirm your search key with ENTER</b> . To view all properties again clear the filter either by clicking on the X icon or using the shortcut <b>SHIFT+BACKSPACE</b> .
	8	Inspector History	Go back and forth between previously inspected items. Either open a drop down menu with a history list to select from or click on the previous/ next buttons (</>).

## Property Tabs

- All properties are **sorted thematically** by tabs.
- The **number and type of tabs vary** depending on the selected item.
- The **Main** tab displays a quick overview of the **most important properties** for the selected item.
- The **All** tab lists **all available properties** in alphabetical order.



Screenshots of all the available tabs sorting the various settings and properties in a Clip Container's Inspector window.

## Change Property Values or Settings

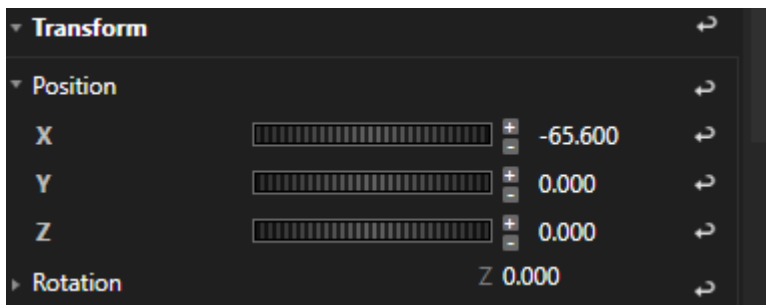
- **Double-click** on a value, enter a number and confirm with "Enter".
- **Reset** a value by clicking on the **return symbol** on the right hand side of the inspector.
- A **little arrow head next to a property name** on the left indicates that there are **child elements** available. Click on the arrow to open.
- **Drag and drop** items like masks or a canvas with your mouse into the checked target field.
- When **multi selecting** items, inspector shows all common properties. Value changes are adopted for all selected elements.



### Mathematical Expressions

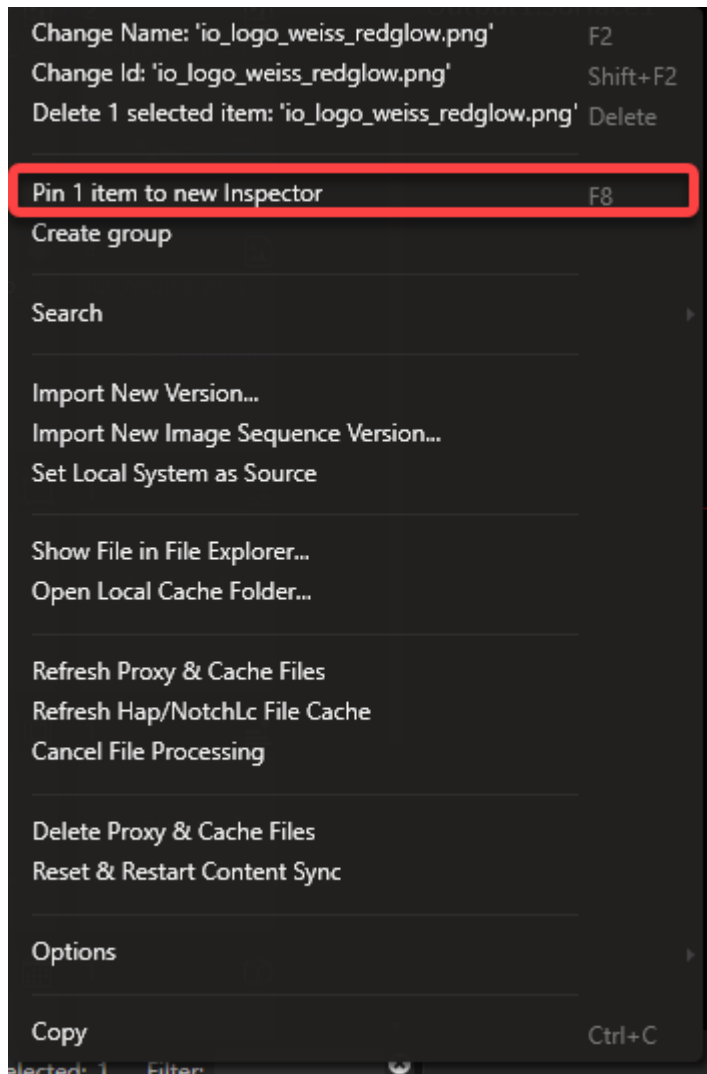
You can use mathematical expression for each value in the inspector.

For instance, enter  $1920*2$  to double your pixel number. Or just add 100 by using  $+100$ .



- For **wheels**: Use **shortkeys** to get more precise results:
  - Hold Alt Key and move wheel with your mouse: increase only whole integer values
  - Hold CTRL Key and move wheel: "Fine Mode"

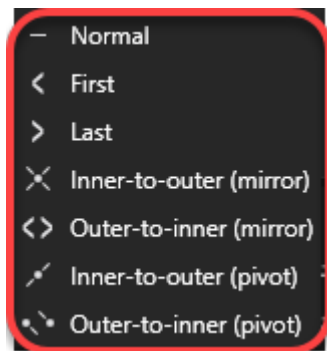
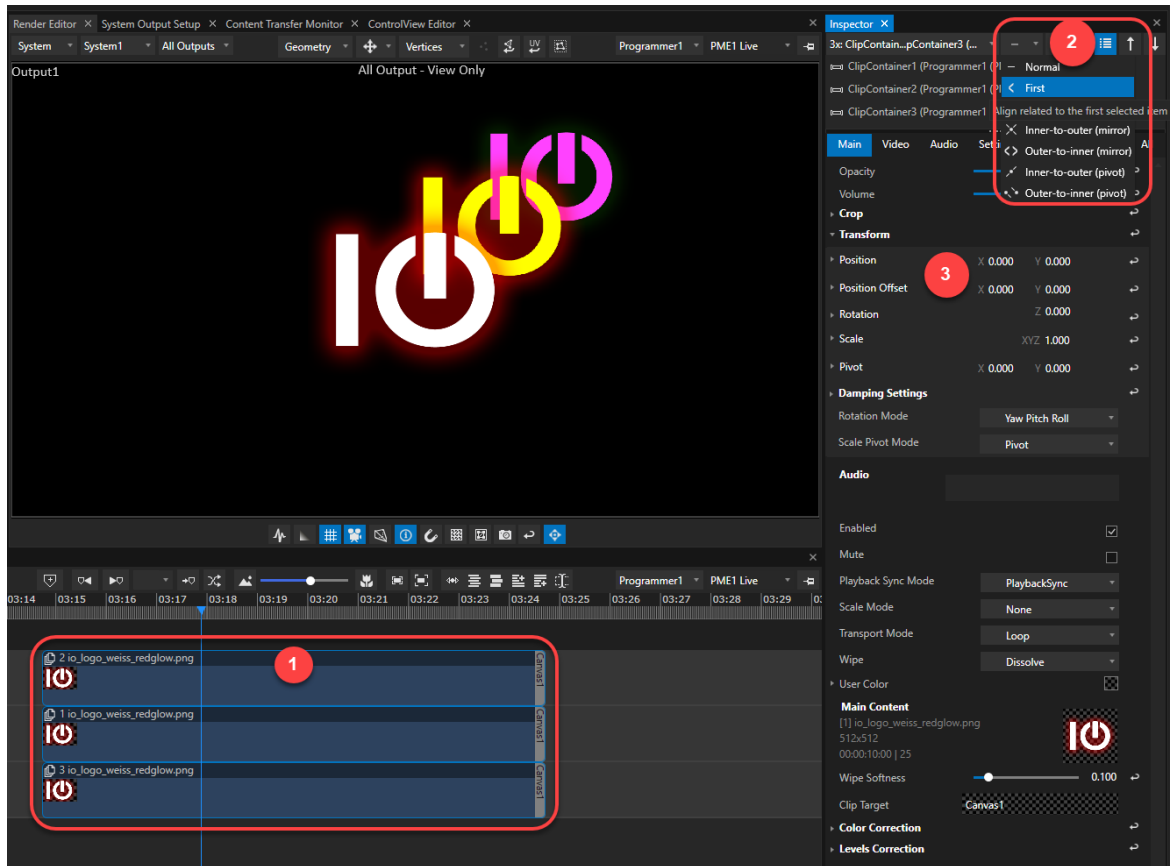
## Pin item to new inspector



- Every item in VERTEX can be accessed with additional and separate inspector windows.
- The item can be optionally **pinned to the new inspector**.
- At times it can be helpful to work with **multiple inspectors to view separate settings distinctly**.

## Align Mode

The Align function offers a quick way to **edit properties** of multiple selections **in relation to one another**. For instance, you can **fan out items on your canvas** evenly and with ease. Here is an example on how to do it:

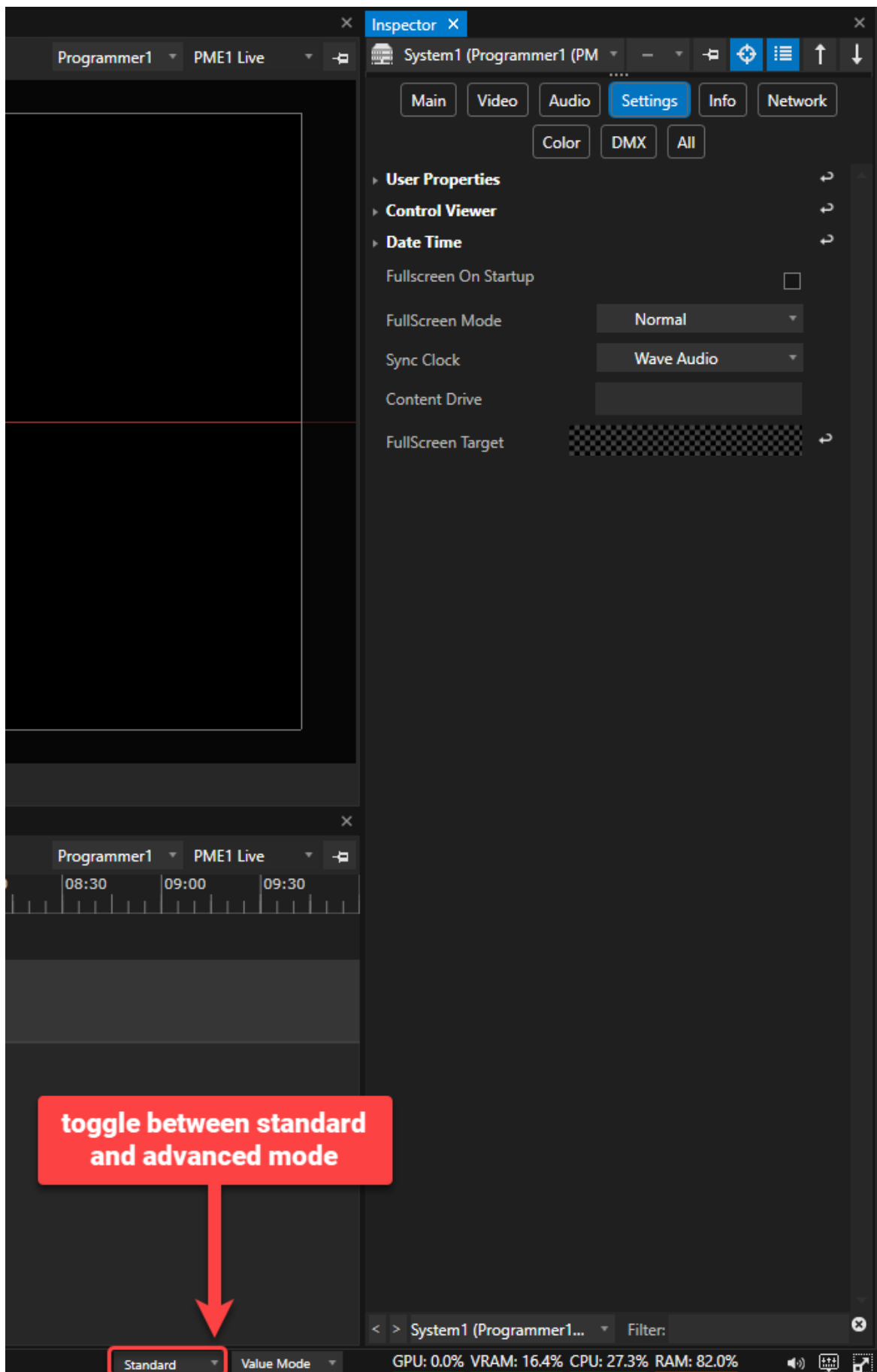


This drop-down menu appears when clicking on the "Align" button (no .2 in UI description).

- 1 Select multiple items such as those three clip containers in the playback editor.
- 2 Select any of the align modes from the drop down in the inspector.
- 3 When changing properties such as position X or Y all three clip containers will be neatly lined up in exact relation to one another.

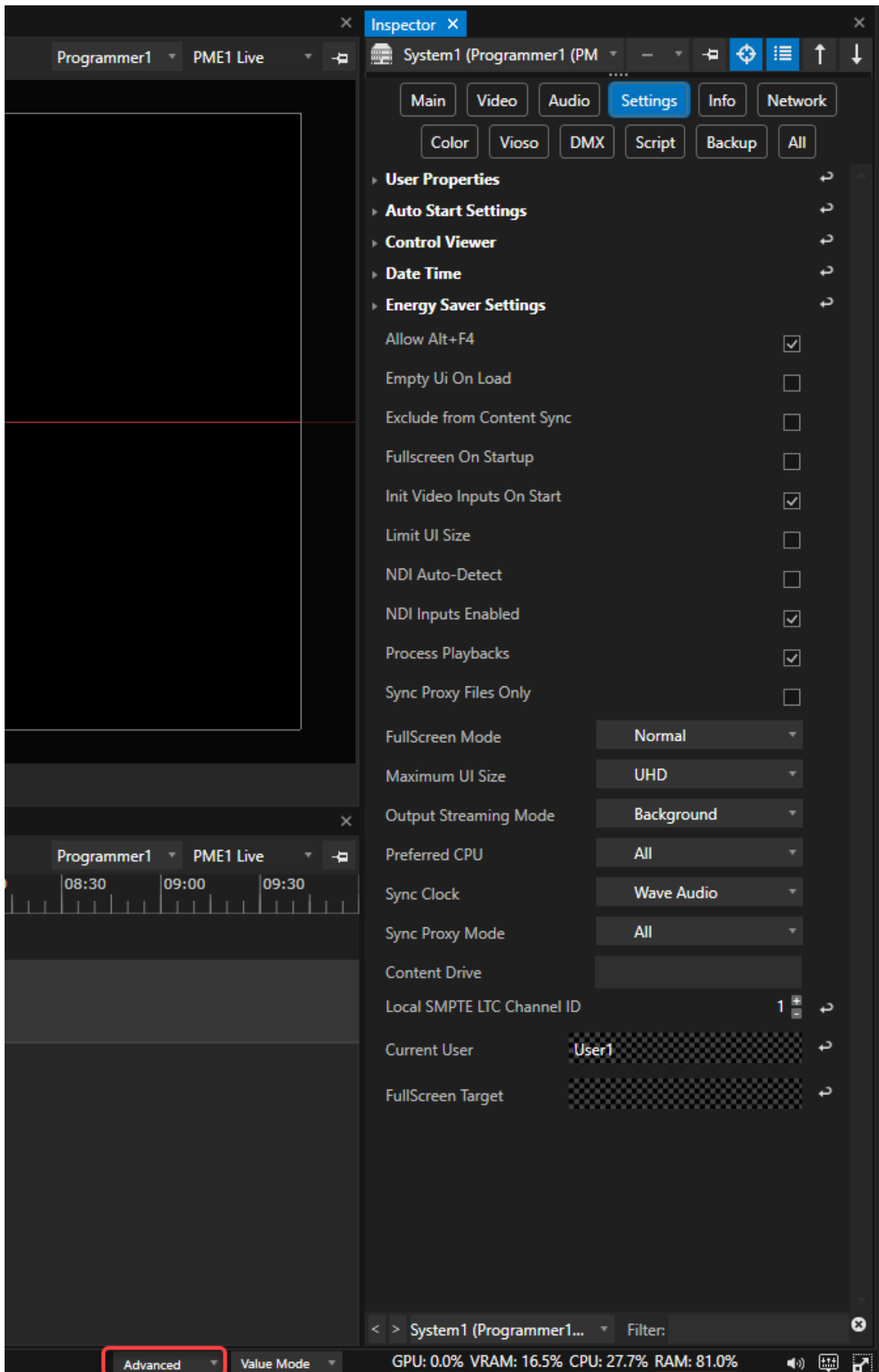
## Inspector Mode: Standard and Advanced

Select one of the inspector's viewing modes from the **drop down menu at the [status bar](#)**. Depending on the inspected item, the number of tabs and the item's properties and options will increase in advanced mode.



This system is being inspected in standard mode.



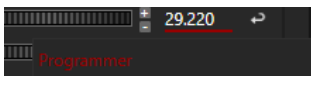
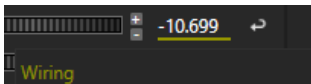
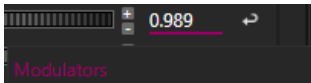
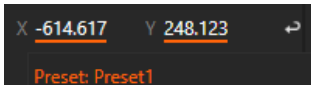
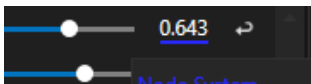
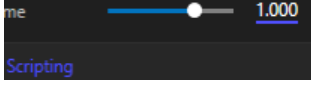

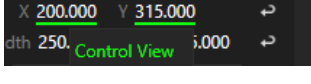


Same system in advanced inspector mode.  
Tab for Vioso calibration settings now is visible.  
The list of system settings has significantly expanded.

## Colors Underlining Value States And Their Meaning

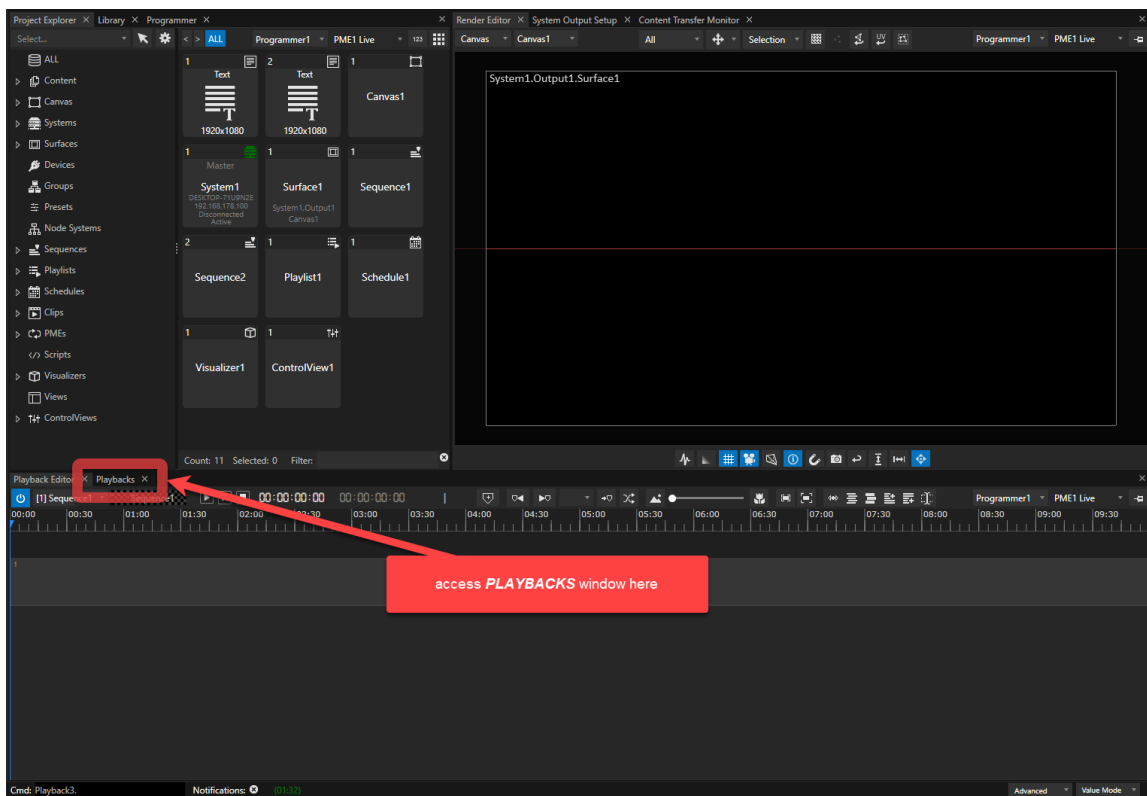
A value state in the inspector can be underlined with a specific color indicating the particular the source of the property value or its changes.

Whenever a parent property entry is underlined, open its drop down to see what child property value is being changed.

	<p><i>Red</i> indicates the source of value state changes is the <b>programmer mode</b>. A darker shade of red indicates multiple programmers.</p>
	<p><i>Yellow</i> indicates the source of value state changes is a <b>wiring</b>. A brighter shade of yellow indicates multiple wirings.</p>
	<p><i>Magenta</i> indicates the source of value state changes is <b>modulators</b> from within a node system such as oscillators, sine waves etc.</p>
	<p><i>Orange</i> indicates the source of value state changes is coming from a <b>preset</b>.</p>
	<p><i>Blue</i> indicates the source of value state changes is a <b>node</b> system.</p>
	<p><i>Dark blue</i> indicates the source of value state changes is a <b>script</b>.</p>
	<p><i>Dark green</i> indicates the source of value state changes is coming from a <b>web view</b>.</p>
	<p><i>Bright green</i> indicates the source of value state changes is coming from a <b>control view</b>.</p>

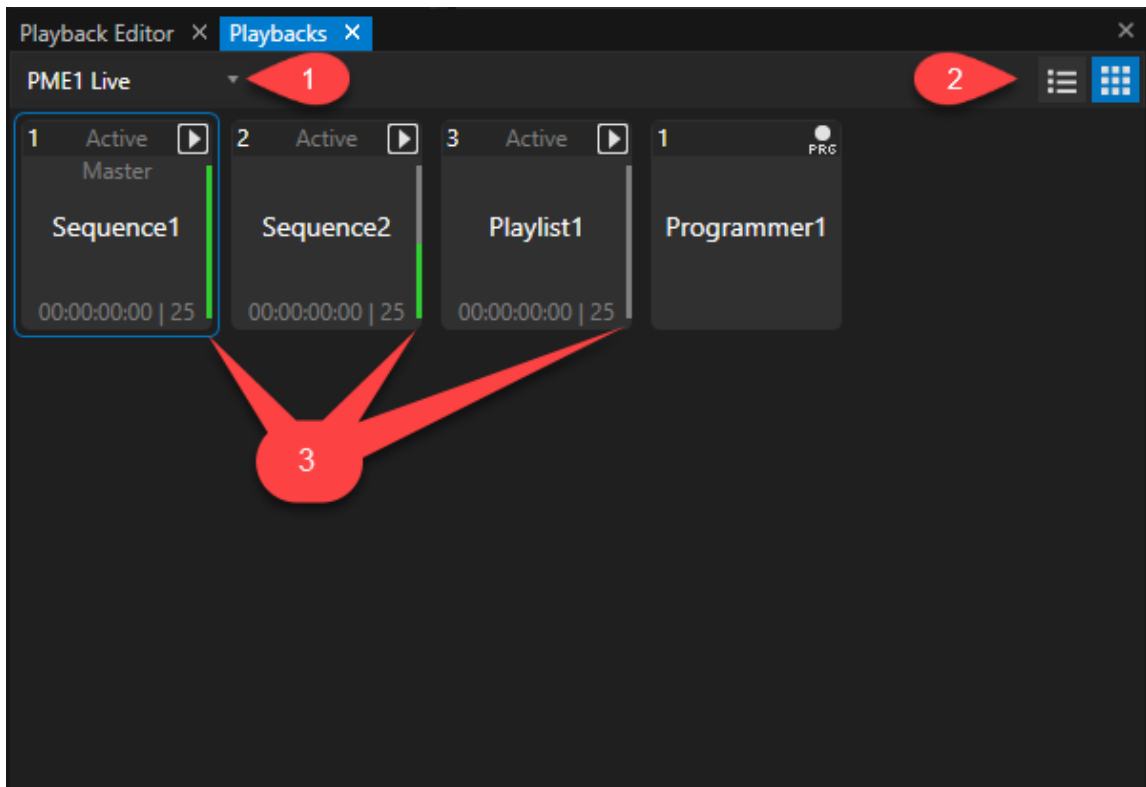
## 5.9.7 Playbacks

- The **Playbacks** window gives you an organized and **quick look on all your playbacks** and their **status**
- There are two views with different focus: A tile view for a fast and good overview and a detailed with with controller buttons for play, pause and cues
- **switch** between **live and preview** playback



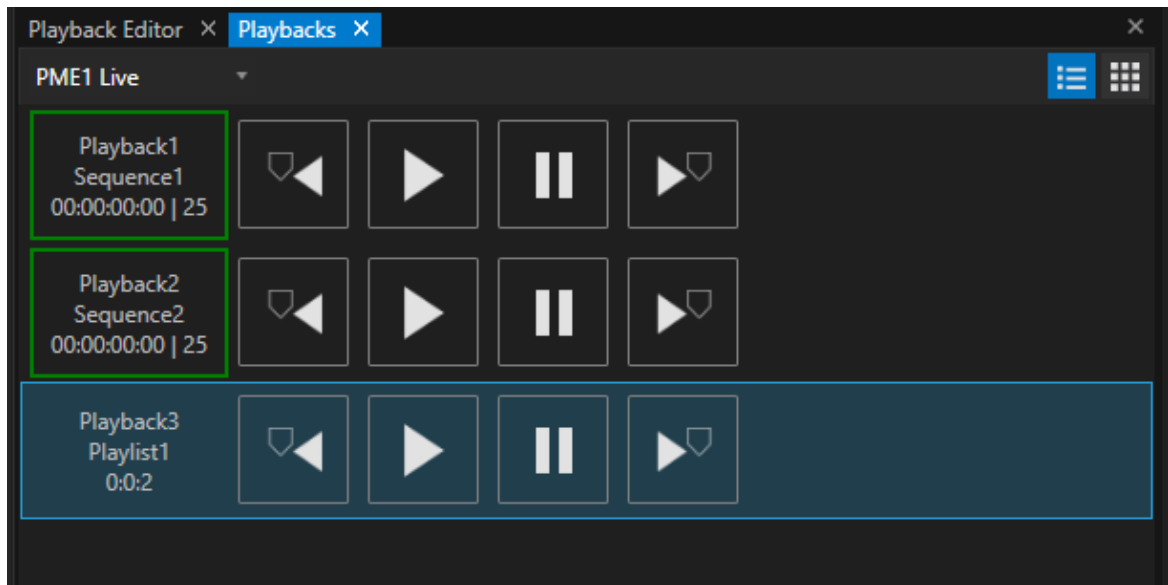
## User Interface

## Tile View



<p><b>1</b></p>	<p><i>PME Selection</i></p>	<p>Select between discrete Playback Mixing Engines: live or preview(s) Default: live</p>
<p><b>2</b></p>	<p><i>Window Layout</i></p>	<p>Select between a tile and a list view</p>
<p><b>3</b></p>	<p><i>Mix Level</i></p>	<p>green status bar showing mix levels Playback1: 100% (default) Playback2: 50 % Playback3: 0 %</p>

## Detailed View



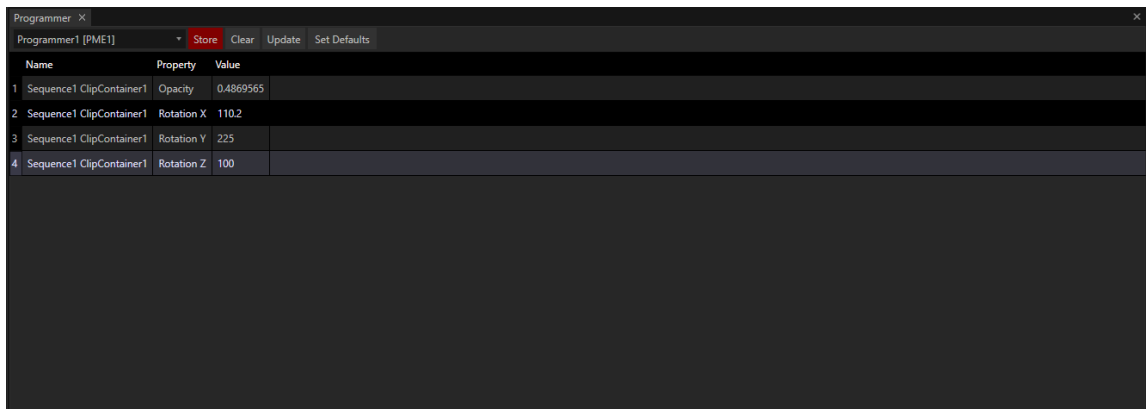
On the left side are all playbacks listed with infos on playback provider, current playhead position and FPS.

On the right are transport controls for each playback: previous cue, play, pause, next cue.

## 5.9.8 Programmer

- The Programmer is a **temporary buffer that caches your changes**. VERTEX hold this changes into Programmer **until you save or clear them**.
- VERTEX has to be set into [Programmer Mode](#) first
- You are able to **use multiple Programmers** into your project

## Workflow



- Changes for Clip Containers and items **are finally saved as keyframes** in the project **only after you click to the save button into Programmer**
- When you store a Programmer list, **VERTEX automatically creates Keyframes**.
- You can **work on a whole scene**, make changes and only **store the final result**. People from the lighting console world should be familiar with this kind of workflow....



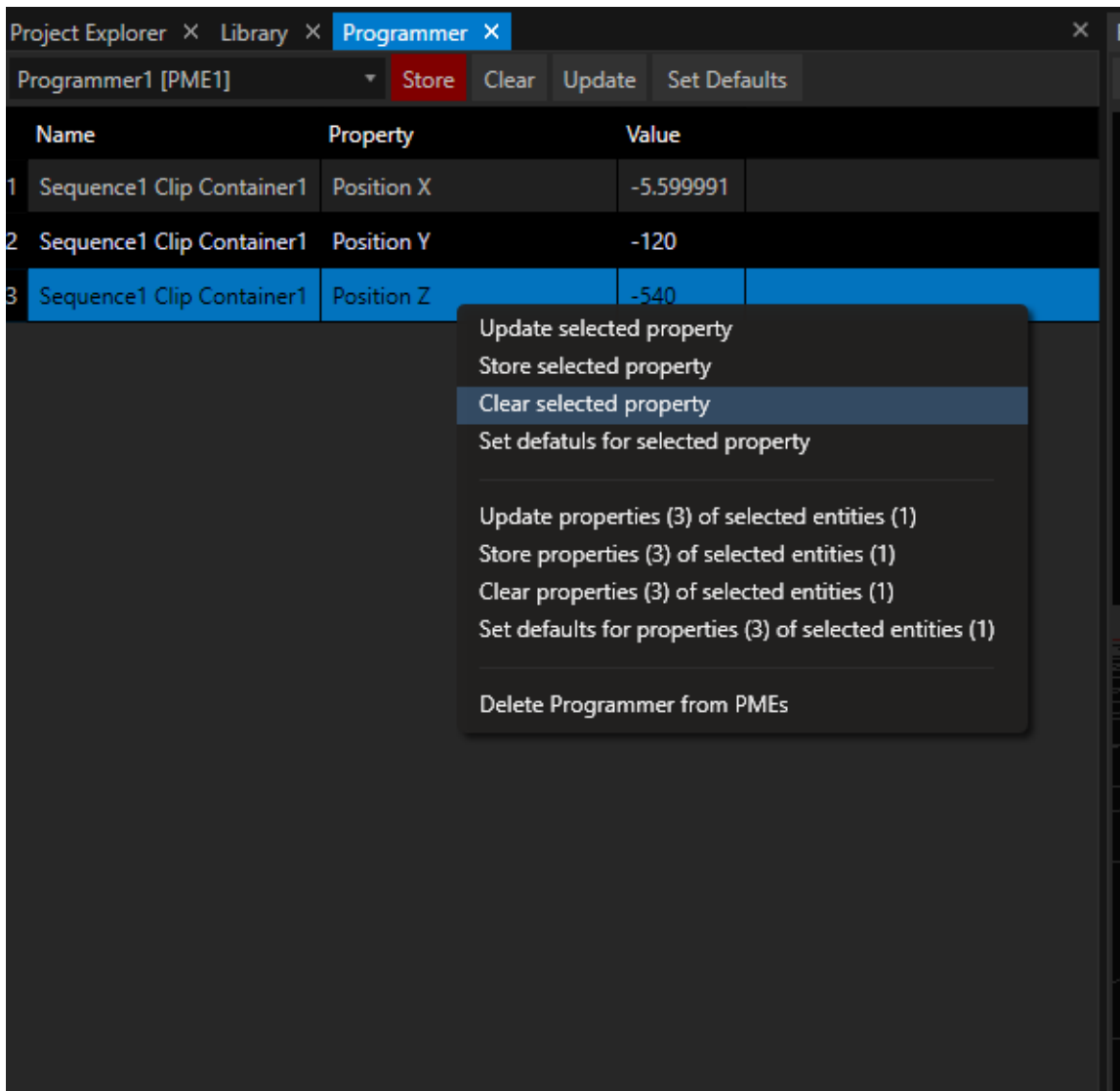
#### **Switch from Value to Programmer Mode first**

VERTEX starts in Value Mode by default. To work with a Programmer, you have to switch VERTEX to "Programmer Mode" first.

The Mode switch is located into Status Bar ( right bottom)

You don't have to decide for one mode. Feel free to switch between value and programmer mode into your project

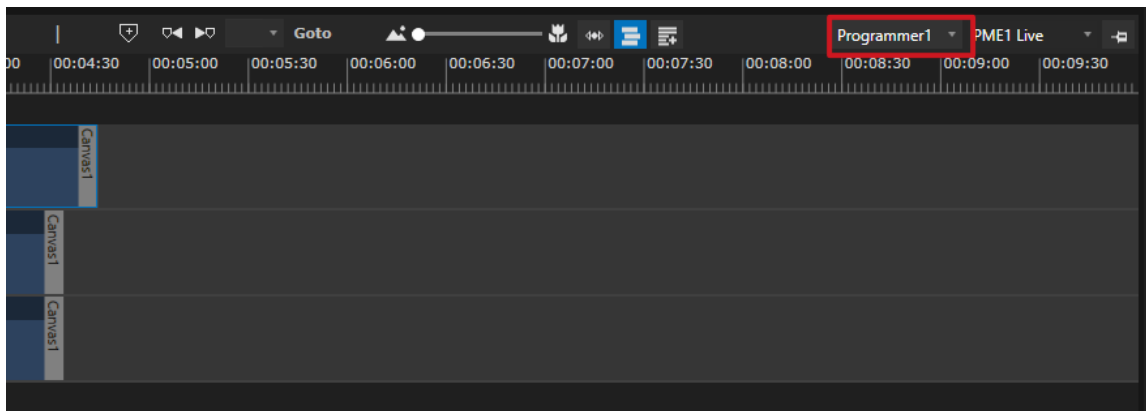
## Clear or Delete Values



- You are able to **clear, revert or just update** single parameters and values into your Scene.
- Just select a Programmer entry and **open the Context Menu** with a right-click

## Select a Programmer to work with

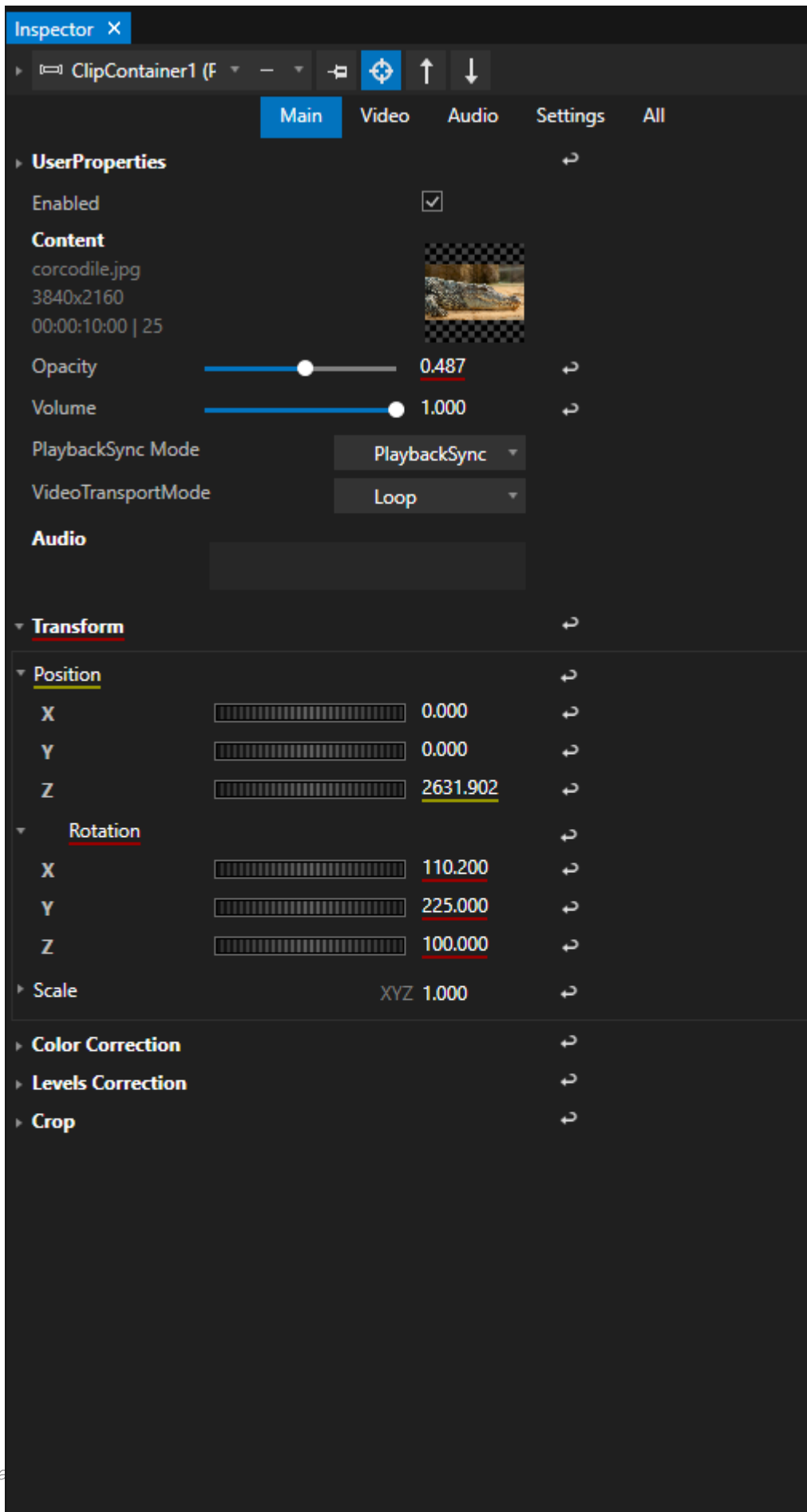
- You are able to work with **multiple Programmers**. For every Editor you can choose a Programmer that VERTEX should use for your changes.



## Values into Inspector

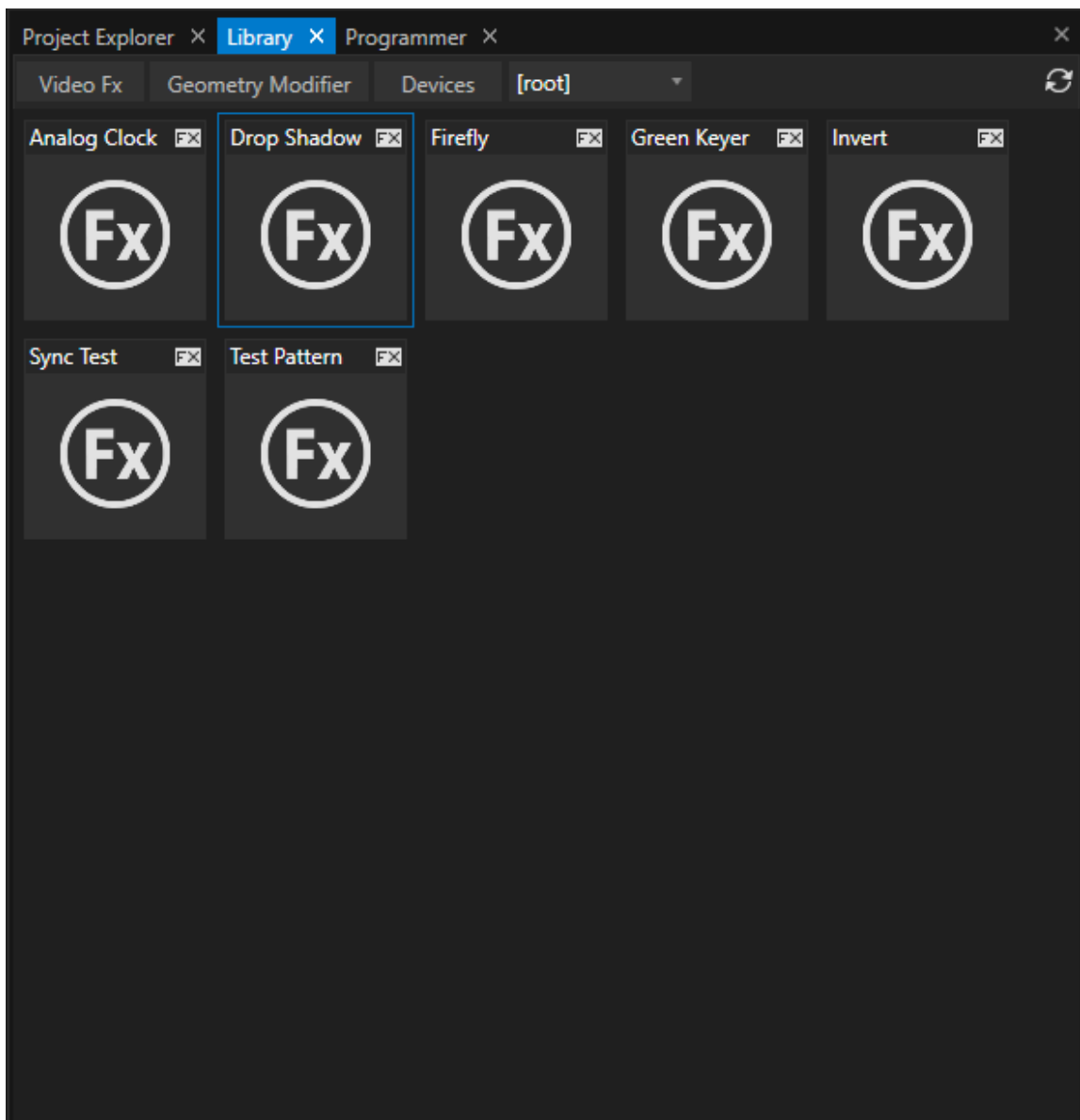
- **Active Values** that are currently used into Programmer are **underlined in red color**
- Every value change into Inspector will **create a new Programmer record** at Playheads position in timeline





## 5.9.9 Library

- The Library Editor gives you access to **Video FX**, **Geometry Modifiers** and **Devices**
- The Library types are **sorted by tabs**
- **Depending on the Library elements type**, **different user actions** are possible.



## Geometry Modifier

- *could be assigned to Clip Containers or Surfaces*

### **Add Geometry Modifiers to an object :**

- *drag it on a selected surfaces or Content directly into the Render Editor*
- *drag it in the Playback Editor on a Clip Container*

## Devices

- *could be added as project elements to the project*

### **Add Devices to the project:**

- *with help of the context menu ( right-click with your mouse on the device)*
- *by drag the device from the Library into another project Explorer window*
- *by drag the device from the Library as Clip Container to the Playback Editor*



### **Directory and Folder for Devices**

*You are able to write your own device templates and add them to the VERTEX library.*

*You have to save your custom devices with the file extension .vxd*

*Devices that are shipped by ioversal are encrypted .vxdx format*

### **Directory and Folder:**

*C:\Users\Public\Documents\ioversal\Vertex\[Vertex Assembly Version]\Devices*

*You are allowed to create own subfolders or copy your device into an already existing subfolder*

## Video FX

- Need a video texture as base - could be added to a clip container with a texture (3d, video, still, text)

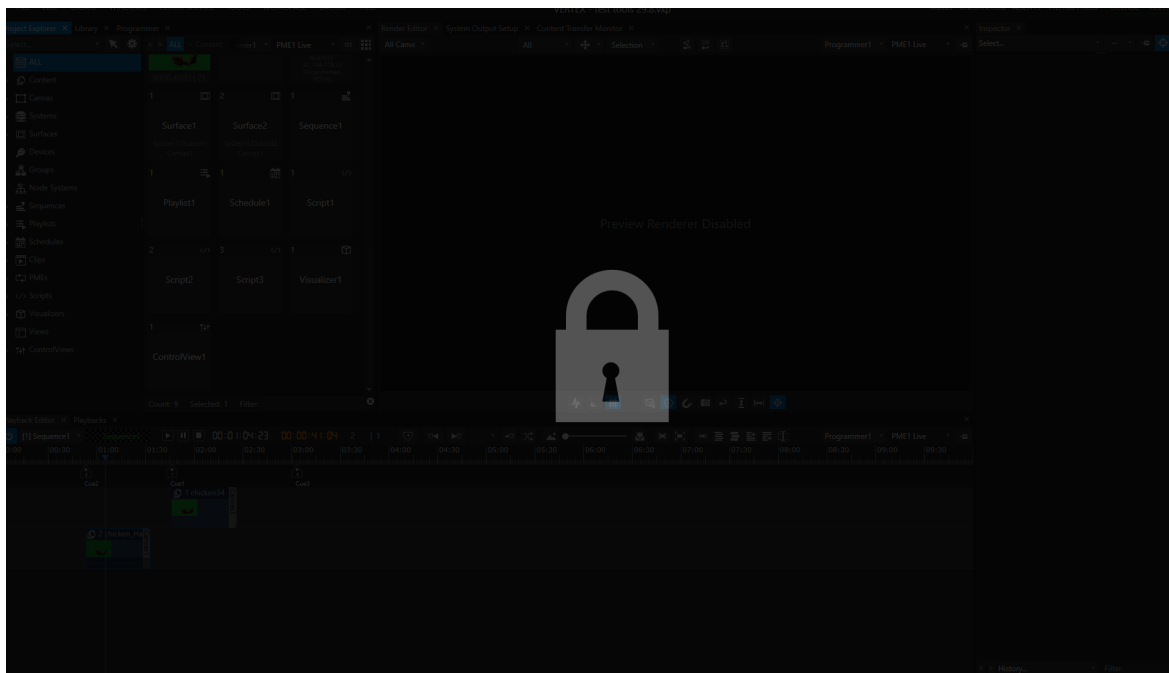
### **Add Video FX**

- drag the FX Shader from Library Editor to a Clip Container in the Playback Editor

## 5.9.10 Lock Screen

To prevent unintentional changes in your project - for instance by accidentally touching mouse or keyboard - you can lock the screen of your user interface.

To do so press the key shortcut CTRL+DELETE. A lock will appear on your screen prohibiting you from interfering with VERTEX until you unlock by pressing CTRL+DELETE again.



## 5.10 Value Mode and Programmer Mode

- **VERTEX combines the advantages of both worlds: Video editing /compositing and lighting console programming.**  
You are able to choose your preferred workflow - or combine both to be faster
- By default VERTEX is set to **Value Mode**: Each setting directly affects an item. Animations could be done with **Keyframes**
- In **Programmer Mode** your settings and changes for items are temporary stored into a Programmer Cache.  
You are able to program whole scenes, make your adjustments there and save only all settings as final result.

### Value Mode

- **Default mode** in VERTEX
- **Global values** for each clip container and item: A value or setting change into Inspector will affect the whole Clip.
- **Animations for Values or Settings** could be done with [Keyframes](#) (or Script Commands)

### Programmer Mode

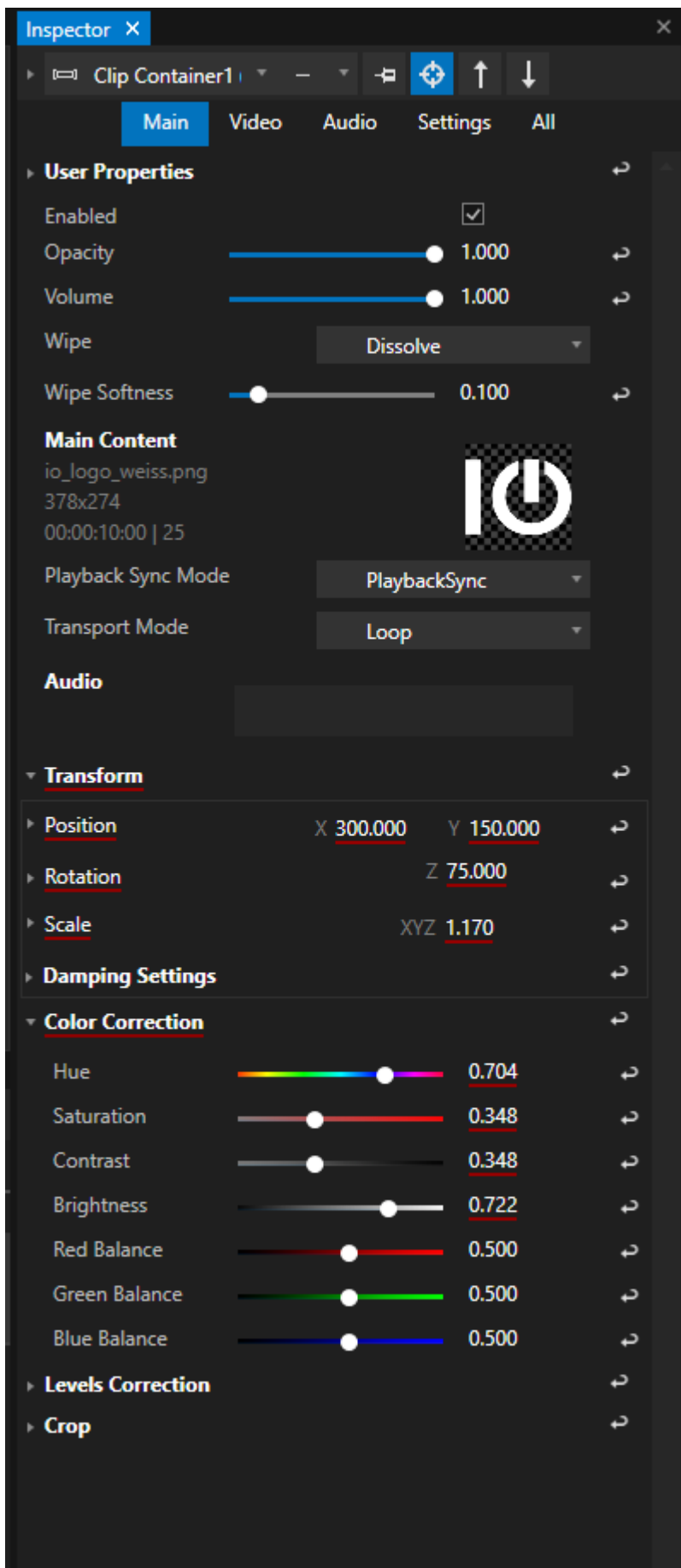
- The [Programmer](#) is a **temporary buffer that caches your changes**. VERTEX **hold** this changes into Programmer **until you save or clear them**.
- Changes for Clip Containers and items **are finally saved as keyframes** in the project **only after you click to the save button into Programmer**
- When you store a Programmer list, **VERTEX automatically creates Keyframes**.
- You are able to **clear, revert or just update** single parameters and values into your Scene.
- You can **work on a whole scene**, make changes and only **store the final result**. People from the lighting console world should be familiar with this kind of workflow...

The screenshot shows a software window titled 'Programmer' with a dropdown menu set to 'Programmer1 [PME1]'. The window contains a table with 10 rows of data. The 'Store' button is highlighted in red. The table columns are 'Name', 'Property', and 'Value'.

	Name	Property	Value
1	Sequence1 Clip Container1	Color Correction Brightness	0.7217391
2	Sequence1 Clip Container1	Color Correction Contrast	0.3478261
3	Sequence1 Clip Container1	Color Correction Hue	0.7043478
4	Sequence1 Clip Container1	Color Correction Saturation	0.3478261
5	Sequence1 Clip Container1	Position X	300
6	Sequence1 Clip Container1	Position Y	150
7	Sequence1 Clip Container1	Rotation Z	75
8	Sequence1 Clip Container1	Scale X	1.17
9	Sequence1 Clip Container1	Scale Y	1.17
10	Sequence1 Clip Container1	Scale Z	1.17

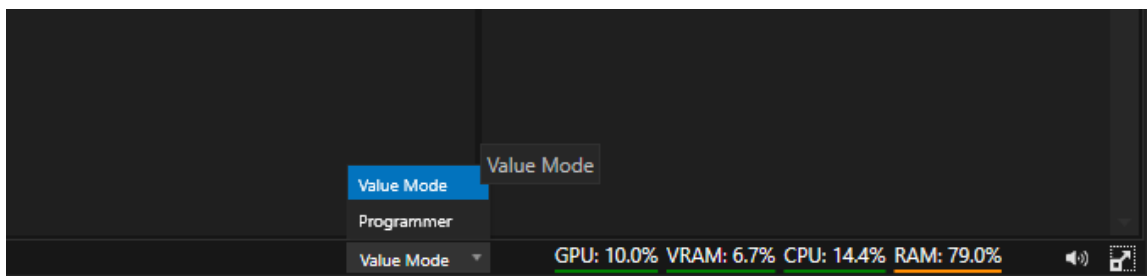
*Programmer with a list of values*

- All Properties and Values that are currently into a Programmer list, are underlined in red Color into Inspector. You get a visual feedback about all your Values that are currently active and into a Programmer



- You are able to create more than 1 Programmer - e.g. When working in a Multi-User Environment or with different Previews

## Switch between Modes



- VERTEX starts in **Value Mode**
- Go to **Status Bar**: on the **right bottom is a drop down** to switch between the modes
- Once switched, VERTEX will work into Programmer Mode until the whole Application is closed.



## *Advanced Features*

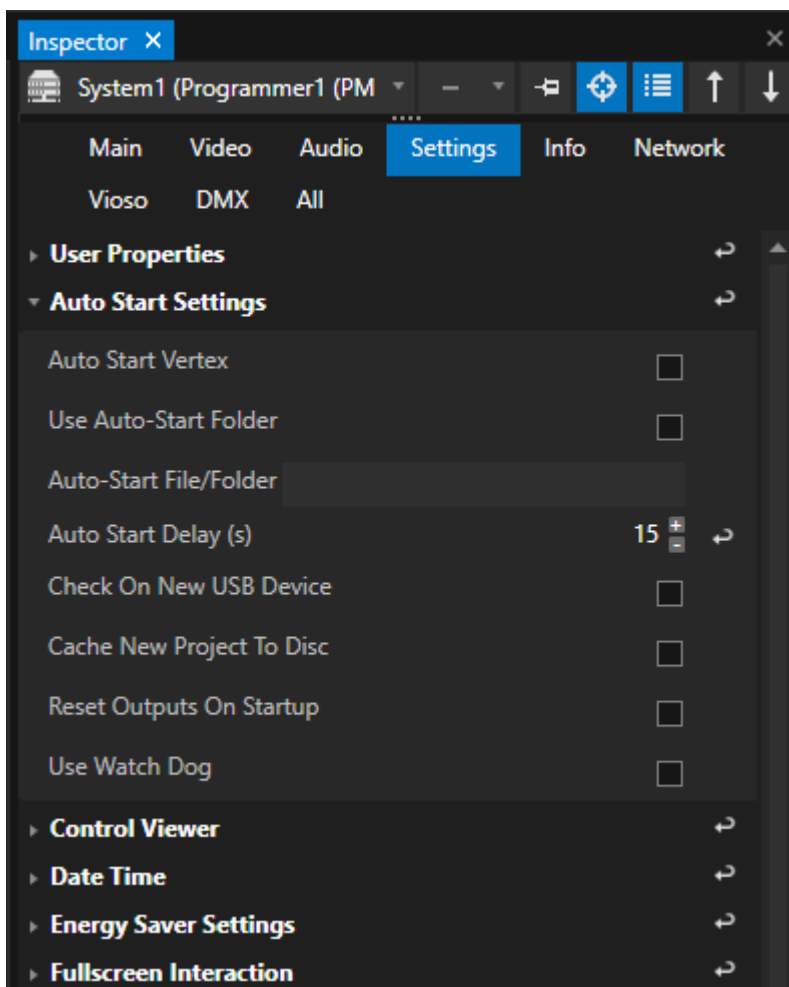
## 6 Advanced Features

*The documentation for the advanced feature-set of VERTEX is in progress and will be updated step by step  
Until then: Please drop us an E-Mail with your "How-to-do-this-in-VERTEX?" question to [support@ioversal.com](mailto:support@ioversal.com)*

## 6.1 Autostart

- VERTEX offers you an easy way to **autostart the application in Windows 10** .
- Select a System and go to **advanced settings in the Inspector** to **pick your Auto Start Settings** in order to automatically start just the application, load a project from a watch folder or load a certain project file.

### Auto Start Settings and Options

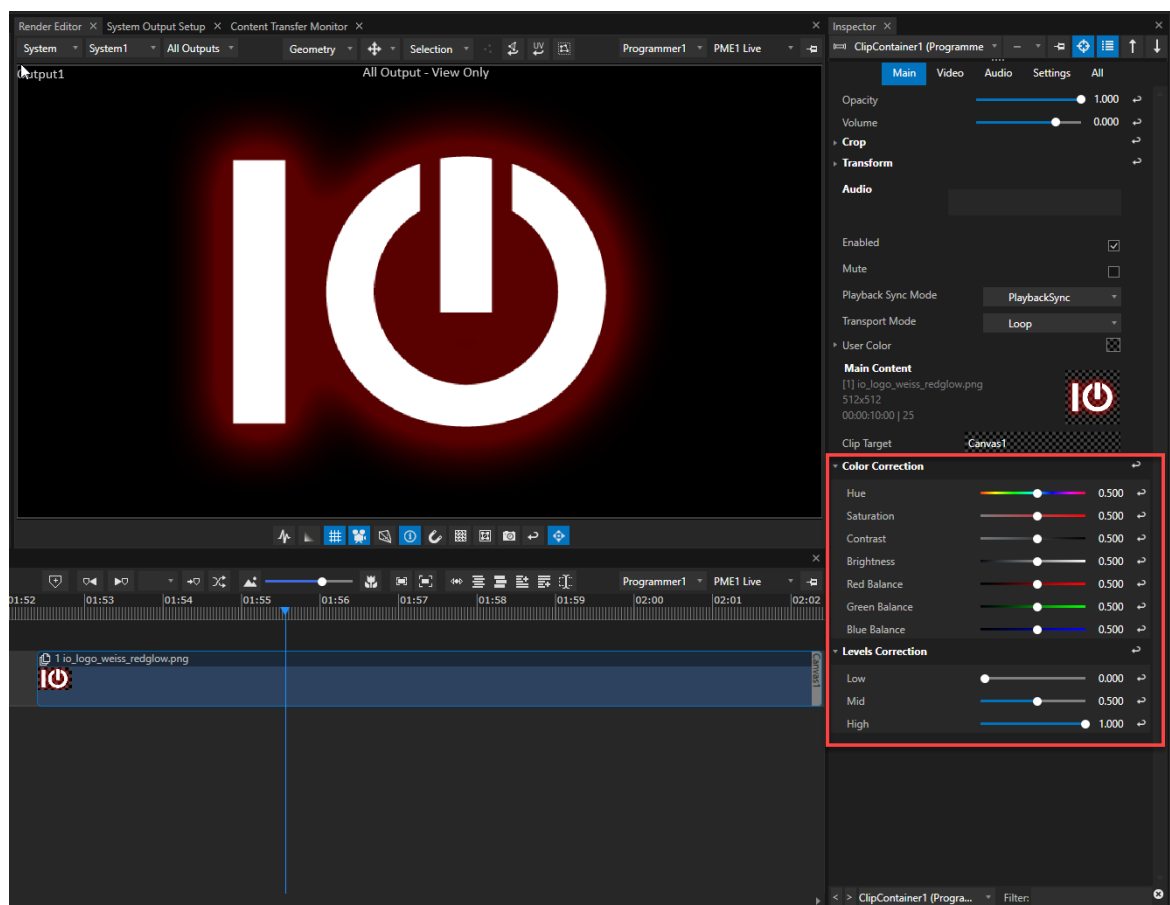


Auto Start VERTEX	Default: Disabled
-------------------	-------------------

	<p>When enabled, VERTEX will automatically start after Windows 10 startup. Use this setting without any specified file or folder in instances where session members are being connected to a master after Windows startup. Specify a folder or project for standalone systems or for a master system ( see below).</p>
<b>Use Auto Start Folder</b>	<p>If enabled, VERTEX will look for .vxp project files in the selected folder after auto start.</p>
<b>Auto Start File/Folder</b>	<p><b>There are 3 options to specify VERTEX behavior and project loading during auto start:</b></p> <p>1) Set a folder: VERTEX will look for a project file in that folder and will load the first project file. This is like a watch folder where VERTEX will always look for a project file.</p> <p>2) Set a path to a specific VERTEX Project File: VERTEX will load the specified .vxp project file. This usage is for master systems starting always with a specific show.</p> <p>3) Set a path to a specific hard drive - for instance: "E:/"          VERTEX will search this external drive for a VERTEX project file to load. For this, please also enable " Check on new USB device".</p>
<b>Auto Start Delay (s)</b>	<p>Sets the time delay between Windows startup and VERTEX's auto start. Default time is set to 15 seconds.</p>
<b>Check on New USB Device</b>	<p>Enable if VERTEX should check for a new external USB devices on auto start to load an existing project from. Please specify the external drives path in the Auto Start File/Folder field.</p>
<b>Cache New Project to Disc</b>	<p>Enable to cache a project from an external drive to disk. Use to prevent your installation or show from being disrupted by someone accidentally un-plugging the external USB drive.          Recommended setting when loading projects from external drives.</p>
<b>Reset Outputs On Startup</b>	<p>If enabled, all outputs and surfaces will be reset to match the current desktop output configuration of your system. Recommended if you run a project on different hardware scenarios.</p>
<b>Use Watch Dog</b>	<p>If enabled, VERTEX will monitor this process and restart if not responding.</p>

## 6.2 Color Correction

- Color correction and color grading are important and powerful tools in post-production video editing and image editing software.
- VERTEX offers those tools for your live production.
- Color correction can be applied to correct underexposure or balance different clip containers in order to avoid color clashes between different clips.

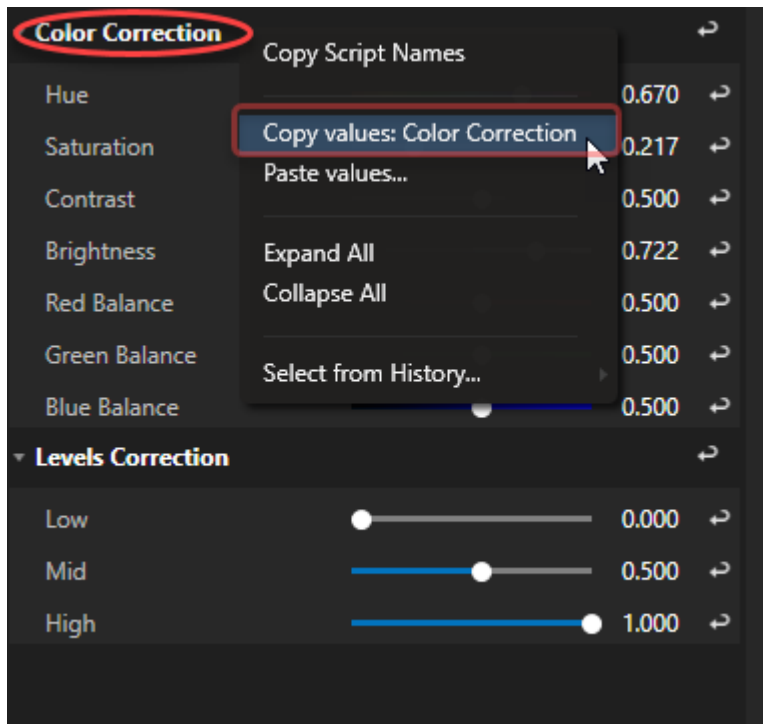


Color correction can be done from the inspector of a clip container:

- **Select a clip container** in your sequence.
- **Go to** the settings for **Color Correction** in **the inspector** at the bottom of the main tab. This can be useful to fix issues with color saturation or skin tones.
- **Underneath** you'll find the settings for **Level Corrections** to for example adjust white balance or fix underexposure.

## Copy and paste correction values with the help of the right-click context menu:

- Open the context menu with a right-click on the parent setting **Color Correction** to **copy all** values.
- Right-click on a single setting (i.e. Saturation) to copy just this setting's value.



*copying entire sets of color correction values can be a real time saver working on multiple clip containers*

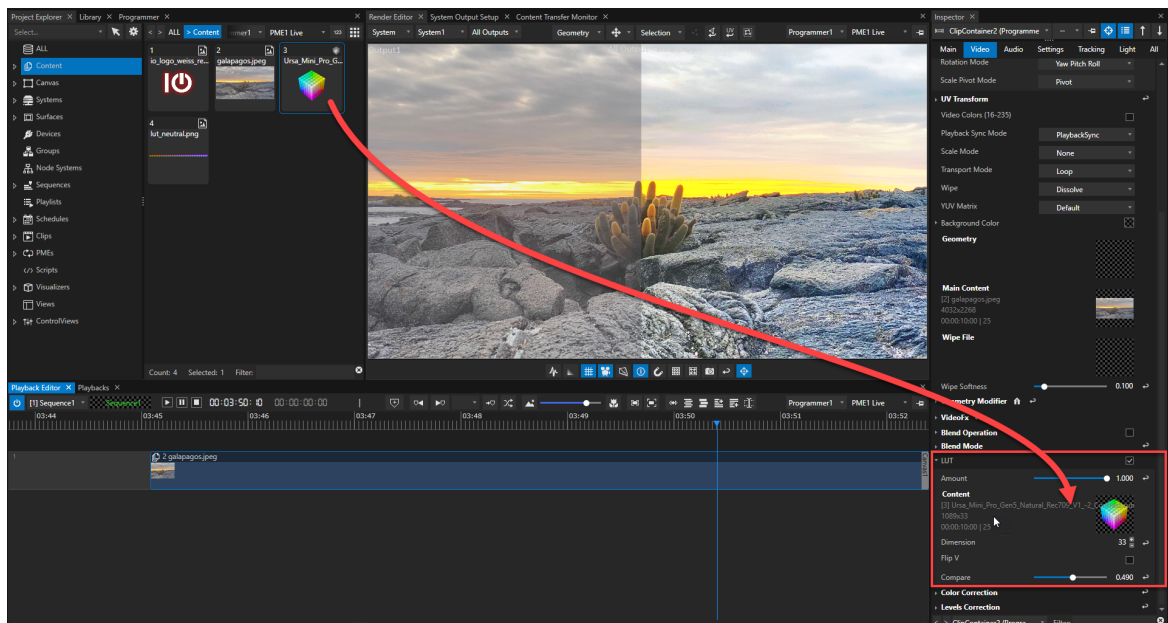
## 6.3 Color Grading

- Whereas color correction fixes issues in your source material, color grading would be the next step in giving your project a unified look in a certain style.
- If you choose to add color grading to your project, it generally should be applied after color correction.

### Color Grading with LUTs

- **LUT** stands for Look-Up Table - **a table of numerical values that transforms an image's contrast and color**. LUTs are commonly used by filmmakers and video editors to efficiently control the aesthetic or look of their footage.
- **Color grading with LUTs can be applied to individual clip containers as well as to entire surfaces** at the output end of your rendering pipeline.
- VERTEX supports 3D LUTs in the file format .cube and custom horizontal PNG strips.

How to apply a LUT to a clip container:

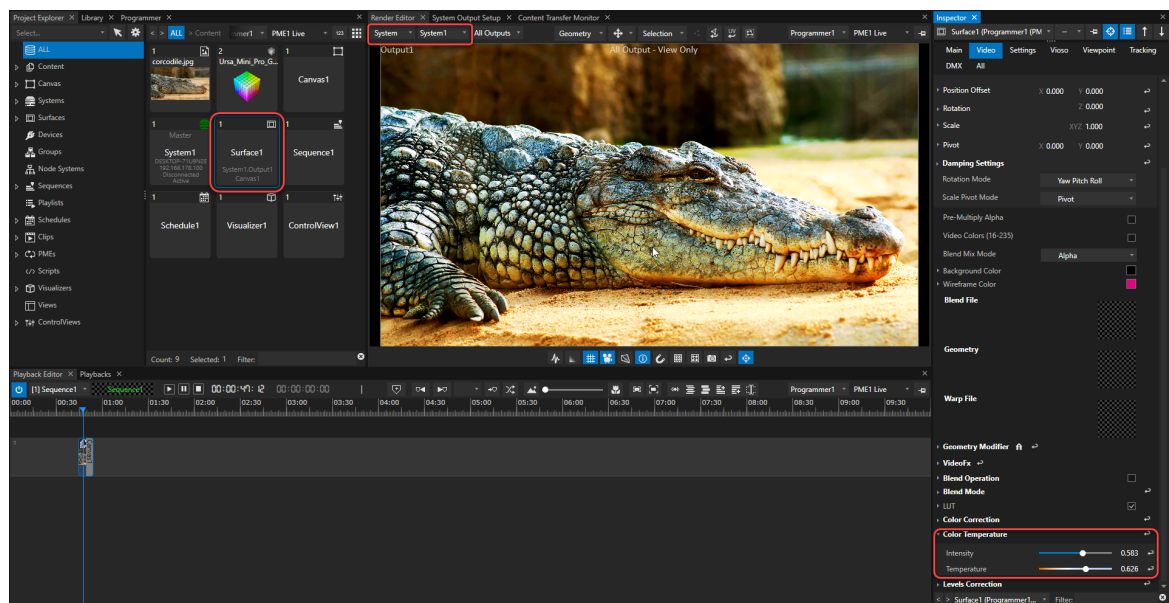


- **Import the LUT** into the content folder of your project explorer (right-click context menu, or MAIN MENU -> FILE -> File... or press CTRL+I).
- **Select the clip container** you want to apply the LUT to and go to the Video tab in the inspector.

- **Drag the LUT** from the project explorer **into the inspector** and drop it into the **field labeled Content** in the LUT settings.
- If the colors show abnormal results try flipping the LUT vertically (Flip V setting).
- Control the blend between the original and the graded colors with the Amount fader.
- Use the Compare fader to split the image into original and graded to compare your results.
- Alternatively, use the LUT enable check box.
- The Dimensions setting allows for values up to 128 when adding a horizontal PNG strip instead of a .cube file.

## Color Grading On Surface Level

- On the top level of VERTEX's rendering engine you can change the color parameters for the entire surface/output.
- First, **change the view mode in your render editor** from **Canvas to System, DMX Mapping or Surface Map**; the render editor won't show you any results in Canvas View if your color editing took place on the surface level.
- **Select a surface** in your project explorer and **go to the Video tab in the inspector**.
- Here you can either do color corrections or apply LUTs for the entire surface as described previously.
- Furthermore you can **pick a color temperature** and adjust its intensity for the surface/output.





## 6.4 Console Layer

- Console Layers gives you the option to control video like a lighting fixture including Iris and shaper
- A Console Layer is **basically independent from a Sequence or a Playlist**. It is only assigned to a Canvas
- with **192 DMX Channels for each Console Layer** you are able to control a bunch of fine adjustments, effects, iris, shaper
- Use **Console Layer as ClipContainer** in your show to **temporary overwrite the global values**

### Channel Map

- Each Console Layer **has 192 DMX-Channels**.
- The channel map could be used as blueprint for a lighting desk fixture/a library fixture

DMX Channel Map for a Console Layer

DMX Address	Parameter Name	Type	Default Value	Note
1	opacity coarse	16bit	0	16bit opacity 1 mode for crossfade wipe modetexture
2	opacity fine	1		
3	opacity wipe mode	1	0	0-255 where 0 no wipe/Crossfade
4	opacity wipe softness	216bit	0	
5	fine			
6	volume coarse	216bit	0	
7	volume fine			
8	Position xpos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
9		xpos fine		
10	ypos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel

11		ypos fine			
12		zpos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
13		zpos fine			
14	Rotation	xrot coarse	216bit	32768	mode crosse fade from absolut to continous value
15		xrot fine			128 = no motion 255=fastForward 0= fastReverse
16		xrot XFade	216bit		2 abs rotation 1 mode 1 rotation speed
17		fine			
18		xrot speed	216bit		
19		fine			
20		yrot coarse	216bit	32768	
21		yrot fine			
22		yrot XFade	216bit		mode crosse fade from absolut to continous value
23		fine			128 = no motion 255=fastForward 0= fastReverse
24	yrot speed	216bit		2 abs rotation 1 mode 1 rotation speed	
25	fine				
26		zrot coarse	216bit	32768	
27		zrot fine			
28		zrot Xfade	216bit		
29		fine			
30		zrot speed	216bit		
31		fine			
32	Scale	xscale	216bit	32768	
33		xscale fine			
34		yscale	216bit	32768	
35		yscale fine			
36		zscale	216bit	32768	

37		zscalefine				
38		OutputFit	1		0= 1to1 Pixel 10-19 Fill, 20-29 Fit, 30-39 Horizontal Fit, 40-49 Vertical Fit	
39	Playback	PlayModes	1		0-7 Pause InFrame, 8-15 Play Once, 16-31 Play Loop, 32-47 Reserved for Ping Pong, 48-63 Play Once Opacity Zero Pause At Current Frame, 64-79 Play Loop Opacity Zero Pause At Current Frame, 80-95 Reserved for Ping Pong Opacity Zero Pause At Current Frame, 122-134 Pause at current Frame, 248-255 Pause OutFrame	
40		Playback Speed	216bit	49152	1=fast reverse 64=normalPlayReverse 128 = Pause 192=Normal Play fwd 254=fast forward	
41		fine			//32769-49152 slow to normal factor 0-1.0 //49152 normal play factor 1 //49153 to 65534 normal to fast fwd 1x-8x	
42		In Frame	216bit	0	65535frames from first frame in	
43		fine				
44		Out Frame	216bit	0	65535frames from last frame in	
45		fine				
46		Gobo MixMode	1	0	Mode to determine if gobo2 is mask or how it is multiplied over gobo1, 0-10 BlackWhite, 11-20 BwInv, 21 -30 Alpha, 31-40 AlphaInv, 41-50 Overlay	
47		Gobo MixLevel	216bit	0	Mix Level for Gobo Blend Mode	
48		fine				
49	Gobo 1	Content Folder	1	0		
50		Content File	1	0		
51		Content UV Mode	1			Texture Wrap Mode: Default, Tile, No Tile, Mirror
52		Content UV Xoffset	216bit	32768		Left to right
53		fine				
54		Content UV Yoffset	216bit	32768		bottom to top
55		fine				
56		Content UV X Scale	216bit	32768		32768 =0 0 = maxTiling 65535 maxScaling
57	fine					
58	Content UV Y	216bit	32768		32768 =0 0 = maxTiling 65535 maxScaling	

		Scale		8	
59		fine			
60	Gobo 2	Mask Folder	1	0	
61		Mask File	1	0	
62		Mask Transform Mode	1		Relative or Absolute Mode - Link Sizing and position to Content
63		Mask UV Mode	1		Texture Wrap Mode:Default,Tile, No Tile, Mirror
64		Mask UV Xoffset	216bit	32768	Left to right
65		fine			
66		Mask UV Yoffset	216bit	32768	bottom to top
67		fine			
68		Mask UV X Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
69		fine			
70	Mask UV Y Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling	
71	fine				
72		Geometry Folder	1		
73		Geometry File	1		
74		res	1		reserved for Geometry playmode
75		res	216bit		reserved for Geometry playspeed
76		res fine			
77		res	216bit		reserved for Geometry InFrame
78		res fine			
79		res	216bit		reserved for Geometry OutFrame
80		res fine			
81	Iris	Iris Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
82		Iris Size	216bit	65535	Default 65535
83		fine			
84		Iris Softness	216bit	3300	Default 3300
85		fine			
86		Iris X	216bit	32768	Default 32768

87				8	
	fine				
88	Iris Y	216bit		32768	Default 32768
89	fine				
90	Shaper	Shaper Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
91		Shaper Softness	216bit	2000	Default 2000
92	fine				
93	SL	216bit			
94	fine				
95	SL Rot	216bit		32768	Default 32768
96	fine				
97	SR	216bit			
98	fine				
99	SR Rot	216bit		32768	Default 32768
100	fine				
101	ST	216bit			
102	fine				
103	ST Rot	216bit		32768	Default 32768
104	fine				
105	SB	216bit			
106	fine				
107	SB Rot	216bit		32768	Default 32768
108	fine				
109	Color Mix	Color Mix Mode	1		Normal, Add, Multiply, InverseMultiply, Highlights, ...
110		Color Mix Level	216bit		
111	fine				
112	Red	216bit			
113	fine				
114	Green	216bit			
115	fine				
116	Blue	216bit			
117	fine				
118	Alpha	216bit			

119		fine			
120	Color Correction	Hue	216bit	32768	
121		fine			
122		Saturation	216bit	32768	
123		fine			
124		Contrast	216bit	32768	
125		fine			
126		Brightness	216bit	32768	
127		fine			
128		Red Balance	216bit	32768	
129		fine			
130		Green Balance	216bit	32768	
131		fine			
132		Blue Balance	216bit	32768	
133		fine			
134		Levels Min	216bit	0	
135		fine			
136		Levels Value	216bit	32768	
137		fine			
138		Levels Max	216bit	65535	
139		fine			
140		BlendMode	1		reserved
141	FX1	FX1 Select	1		
142		FX1 Mix	216bit		
143		fine			
144		FX1 C1	216bit		
145		fine			

146		FX1 C2	216bit			
147		fine				
148		FX1 C3	216bit			
149		fine				
150		FX1 C4	216bit			
151		fine				
152		FX1 C5	216bit			
153		fine				
154	FX2	FX2 Select	1			
155		FX2 Mix	216bit			
156		fine				
157		FX2 C1	216bit			
158		fine				
159		FX2 C2	216bit			
160		fine				
161		FX2 C3	216bit			
162		fine				
163		FX2 C4	216bit			
164		fine				
165		FX2 C5	216bit			
166		fine				
167	FX3	FX3 Select	1			
168		FX3 Mix				
169		fine	216bit			
170		FX3 C1	216bit			
171		fine				
172		FX3 C2	216bit			
173		fine				
174		FX3 C3	216bit			
175		fine				
176		FX3 C4	216bit			
177		fine				
178		FX3 C5	216bit			
179		fine				
180	FX4	FX4 Select	1			
181		FX4 Mix	216bit			
182		fine				
183		FX4 C1	216bit			
184		fine				
185		FX4 C2	216bit			
186		fine				
187		FX4 C3	216bit			
188		fine				
189		FX4 C4	216bit			

190	fine				
191	FX4 C5	216bit			
192	fine				

**i** **ChamSys**  
 For ChamSys consoles, the maximum number of DMX channels of a library fixture is limited. Therefore a VERTEX Console Layer is splitted into 2 different ChamSys fixtures. Please see the ChamSys channel map below.  
 When using Console Layers with ChamSys, please switch the DMX mode of a Console Layer (Inspector Tab "DMX", Property "DMX Mode") from "Standard" to "ChamSys".

DMX-Channel Map for a Console Layer and ChamSys Consoles

58 DM X Address	Parameter Name	Type	Default Value	Note	
Fixture 1 A	1 1	opacity coarse	116bit	0	16bit opacity 1 mode for crossfade wipe modetexture
	2	opacity fine	1		
	1 3	opacity wipe mode	1	0	0-255 where 0 no wipe/Crossfade
	1 4	opacity wipe softness	216bit	0	
	5	fine			
	1 6	volume coarse	216bit	0	
	7	volume fine			
	1 8	Position xpos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
	9	xpos fine			
	1 10	ypos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
	11	ypos fine			
	1 12	zpos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel



13		zpos fine				
1 14	Rotation	xrot coarse	216bit	327 68	mode	cross fade from absolut to continous value
15		xrot fine				128 = no motion 255=fastForward 0=fastReverse
1 16		yrot coarse	216bit	327 68		
17		yrot fine				
1 18		zrot coarse	216bit	327 68		
19		zrot fine				
1 20	Scale	xscale	216bit	327 68		
21		xscale fine				
1 22		yscale	216bit	327 68		
23		yscale fine				
1 24		zscale	216bit	327 68		
25		zscale fine				
1 26		OutputFit	1			0= 1to1 Pixel 10-19 Fill, 20-29 Fit, 30-39 Horizontal Fit, 40-49 Vertical Fit
1 27	Playback	PlayModes	1			0-7 Pause InFrame, 8-15 Play Once, 16-31 Play Loop, 32-47 Reserved for Ping Pong, 48-63 Play Once Opacity Zero Pause At Current Frame, 64-79 Play Loop Opacity Zero Pause At Current Frame, 80-95 Reserved for Ping Pong Opacity Zero Pause At Current Frame, 122-134 Pause at current Frame, 248-255 Pause OutFrame
1 28		Playback Speed	216bit	491 52		1=fast reverse 64=normalPlayReverse 128 = Pause 192=Normal Play fwd 254=fast forward
29		fine				//32769-49152 slow to normal factor 0-1.0 //49152 normal play factor 1 //49153 to 65534 normal to fast fwd 1x-8x
1 30		In Frame	216bit	0		65535frames from first frame in
31		fine				
1 32		Out Frame	216bit	0		65535frames from last frame in
33		fine				
1 34		Gobo MixMode	1		0	Mode to determine if gobo2 is mask or how it is multiplied over gobo1, 0-10 BlackWhite,

					11-20 BwInv,21 -30 Alpha,31-40 AlphaInv,41-50 Overlay
1	35	Gobo MixLevel	216bit	0	Mix Level for Gobo Blend Mode
	36	fine			
1	37	Gobo 1 Content Folder	1	0	
1	38	Content File	1	0	
1	39	Content UV Mode	1		Texture Wrap Mode: Default, Tile, No Tile, Mirror
1	40	Content UV Xoffset	216bit	32768	Left to right
	41	fine			
1	42	Content UV Yoffset	216bit	32768	bottom to top
	43	fine			
1	44	Content UV X Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
	45	fine			
1	46	Content UV Y Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
	47	fine			
1	48	Gobo 2 Mask Folder	1	0	
1	49	Mask File	1	0	
1	50	Mask Transform Mode	1		Relative or Absolute Mode - Link Sizing and position to Content
1	51	Mask UV Mode	1		Texture Wrap Mode:Default,Tile, No Tile, Mirror
1	52	Mask UV Xoffset	216bit	32768	Left to right
	53	fine			
1	54	Mask UV Yoffset	216bit	32768	bottom to top
	55	fine			
1	56	Mask UV X Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
	57	fine			

1	58	Mask UV	216bit	327	32768 =0 0 = maxTiling 65535 maxScaling
	59	Y Scale		68	
		fine			
1	60	Geometry Folder	1		
1	61	Geometry File	1		
1	62	Iris Iris Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
1	63	Iris Iris Size	216bit	655	Default 65535
	64	fine		35	
1	65	Iris Softness	216bit	330	Default 3300
	66	fine		0	
1	67	Iris X	216bit	327	Default 32768
	68	fine		68	
1	69	Iris Y	216bit	327	Default 32768
	70	fine		68	
1	71	Shaper Shaper Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
1	72	Shaper Shaper Softness	216bit	200	Default 2000
	73	fine		0	
1	74	SL	216bit		
	75	fine			
1	76	SL Rot	216bit	327	Default 32768
	77	fine		68	
1	78	SR	216bit		
	79	fine			
1	80	SR Rot	216bit	327	Default 32768
	81	fine		68	
1	82	ST	216bit		
	83	fine			
1	84	ST Rot	216bit	327	Default 32768
	85	fine		68	
1	86	SB	216bit		
	87	fine			
1	88	SB Rot	216bit	327	Default 32768
	89	fine		68	

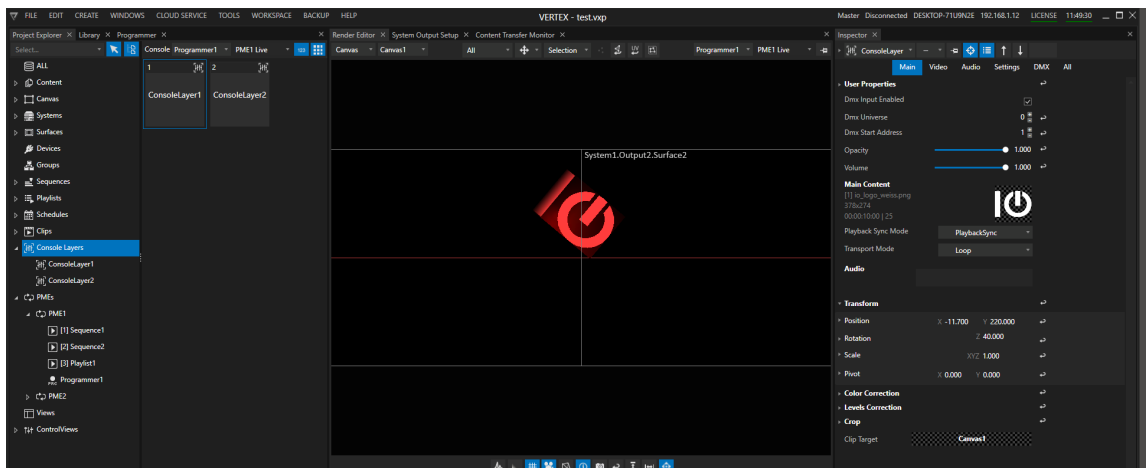
1 90	Color Mix	Color Mix Mode	1			Normal, Add, Multiply, InverseMultiply, Highlights, ...
1 91		Color Mix Level	216bit			
92		fine				
1 93		Red	216bit			
94		fine				
1 95		Green	216bit			
96		fine				
1 97		Blue	216bit			
98		fine				
1 99		Alpha	216bit			
100		fine				
Fixt ure 1 B	1 1	Color Correction	Hue	216bit	327 68	
	2		fine			
	1 3		Saturati on	216bit	327 68	
	4		fine			
	1 5		Contrast	216bit	327 68	
	6		fine			
	1 7		Brightne ss	216bit	327 68	
	8		fine			
	1 9		Red Balance	216bit	327 68	
	10		fine			
	1 11		Green Balance	216bit	327 68	
	12		fine			
	1 13		Blue Blanace	216bit	327 68	
	14		fine			
	1 15		Levels Min	216bit	0	
	16		fine			
	1 17		Levels Value	216bit	327 68	
	18		fine			
	1 19		Levels Max	216bit	655 35	
	20		fine			

1	21	BlendM ode	1			reserved
1	22	FX1 FX1 Select	1			
1	23	FX1 Mix	216bit			
	24	fine				
1	25	FX1 C1	216bit			
	26	fine				
1	27	FX1 C2	216bit			
	28	fine				
1	29	FX1 C3	216bit			
	30	fine				
1	31	FX1 C4	216bit			
	32	fine				
1	33	FX1 C5	216bit			
	34	fine				
1	35	FX2 FX2 Select	1			
1	36	FX2 Mix	216bit			
	37	fine				
1	38	FX2 C1	216bit			
	39	fine				
1	40	FX2 C2	216bit			
	41	fine				
1	42	FX2 C3	216bit			
	43	fine				
1	44	FX2 C4	216bit			
	45	fine				
1	46	FX2 C5	216bit			
	47	fine				
1	48	FX3 FX3 Select	1			
1	49	FX3 Mix				
	50	fine	216bit			
1	51	FX3 C1	216bit			
	52	fine				
1	53	FX3 C2	216bit			
	54	fine				
1	55	FX3 C3	216bit			
	56	fine				
1	57	FX3 C4	216bit			
	58	fine				
1	59	FX3 C5	216bit			
	60	fine				
1	61	FX4 FX4 Select	1			
1	62	FX4 Mix	216bit			

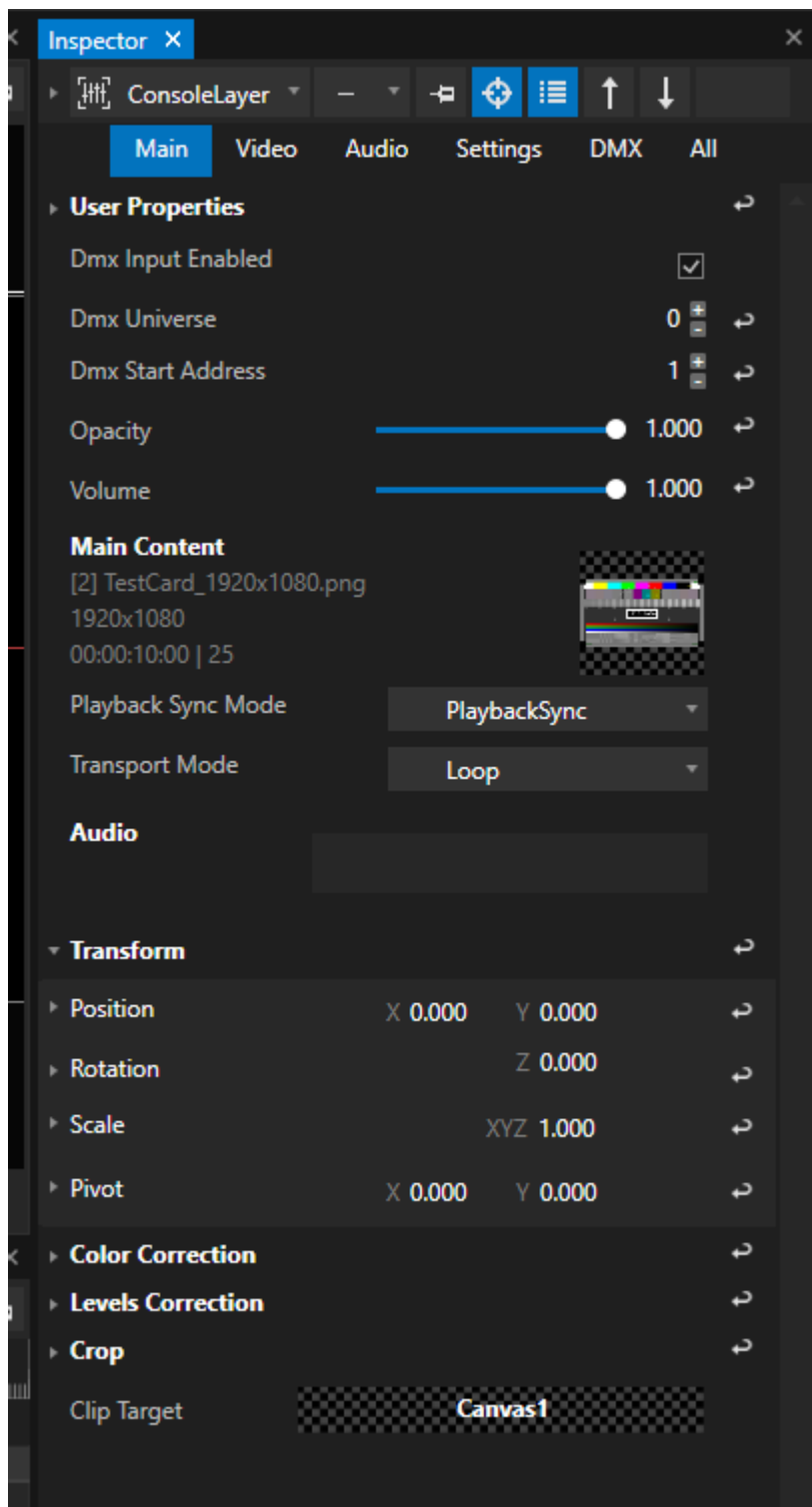
63	fine				
1 64	FX4 C1	216bit			
65	fine				
1 66	FX4 C2	216bit			
67	fine				
1 68	FX4 C3	216bit			
69	fine				
1 70	FX4 C4	216bit			
71	fine				
1 72	FX4 C5	216bit			
73	fine				

39

## Working with a Console Layer



- Use the "Create " tab of the Main Menu to create a new Console Layer
- Right-Click on the ConsoleLayer section into Project Explorer and use the Context Menu the create a new one



- Select Console Layer into **Inspector** and do **initial settings**:
- Set the **target Canvas** on which the Content should be rendered

- Set the **DMX universe** and **start address** if the console layer should be controlled by a lighting desk. Please do a **DMX-Routing** first. Check the **Channel Map** for a Console Layer in detail and learn more about which DMX-512 channel controls which parameter



**Unique start address for each Console Layer**

please keep in mind to change the start addresses for every console layer.

To be control separately, each console layer must have it's unique DMX start address

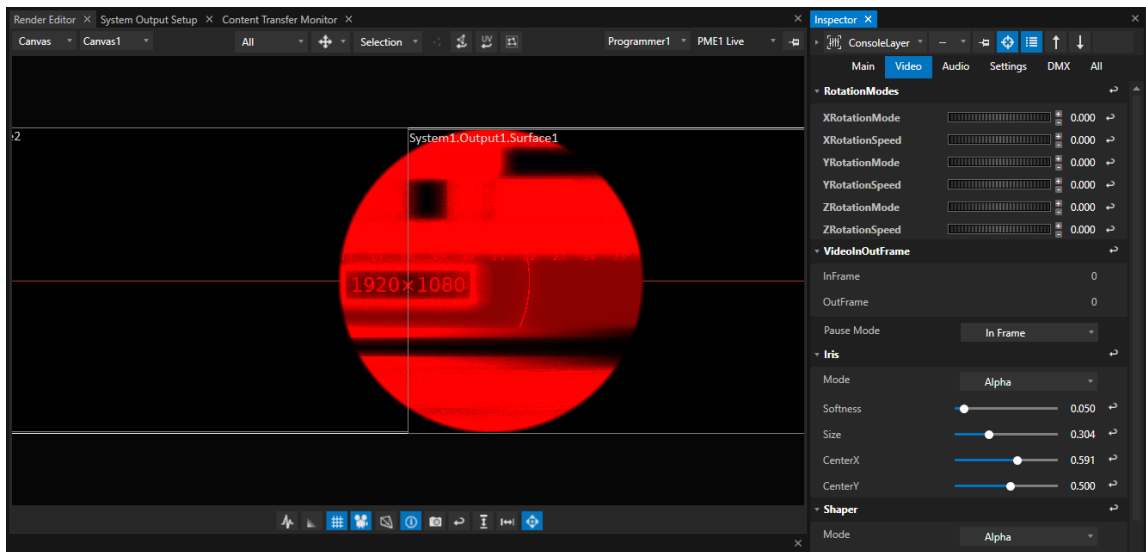
- If you want to use DMX-Values to select Content from Project Explorer like a Gobo, set **DMX File and Folder IDs** for your Content.



**Channel Map: Gobo 1 and Gobo 2**

Gobo 1 in Channel Map is Property "Main Content" (Video tab in the Inspector)

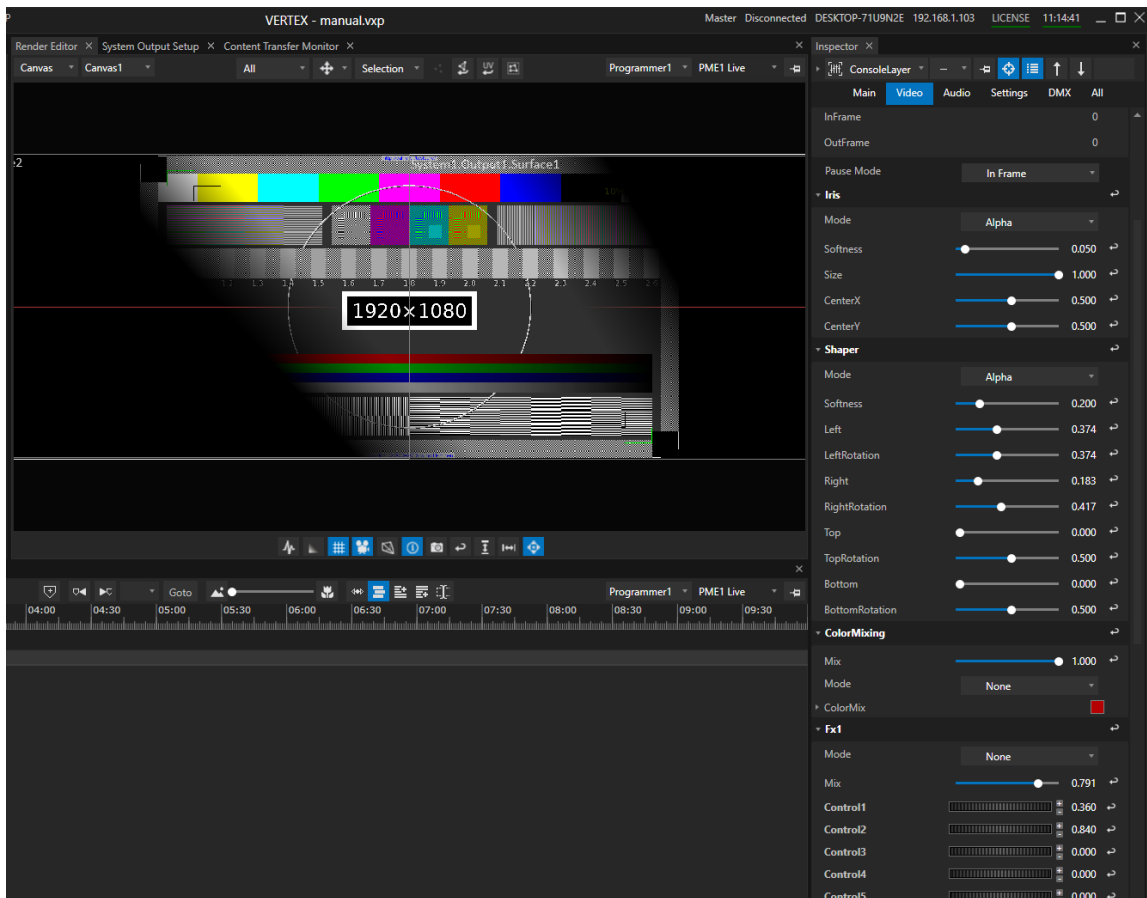
Gobo 2 in Channel Map is Property "Mask" ( Video tab in the Inspector)



Iris and Color Mixing on a Example Content

- Control Console Layer by external DMX or use the Inspector to do settings for all parameters





Shaper for a Test pattern Content

## DMX Folder and File ID

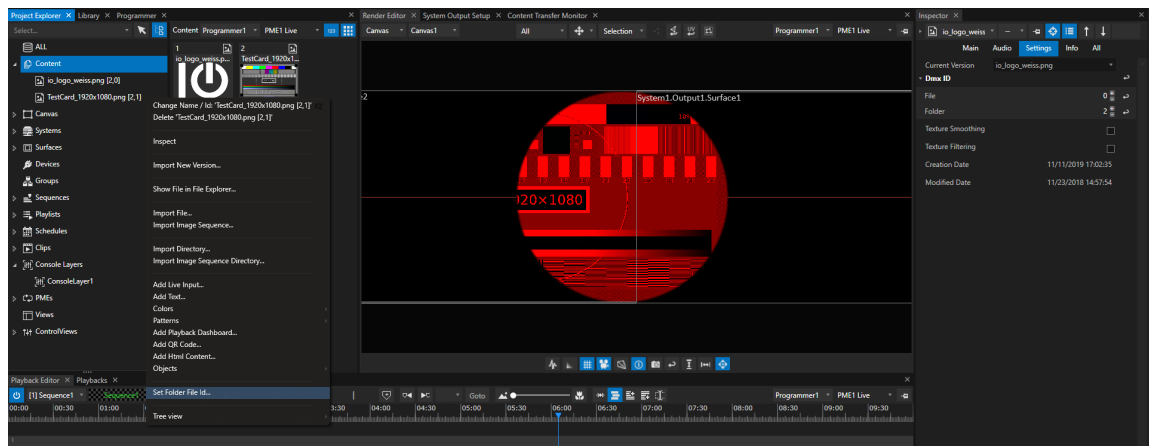
- Each Console Layer is able to host Video Content and an Content Item as a Mask.
- With the concept of folder and file IDs you are able to select your content based on a DMX Value between 0 and 255.
- Each Console Layer has a DMX Channel/Address for Content Folder ID and Content File ID (Channels 49/50) and two Channels/Addresses for Mask Content Folder and Mask Content File (Channels 60/61). Please also compare with the full [channel map of a console Layer](#)

### Example:

- Your Console Layer into VERTEX has DMX Start Address 1 (For universe routing, please read topic [DMX-Routing](#) before)
- Your lighting desk has a library element for a VERTEX Console Layer, also on Start Address 1
- you want to select a Content Item from the VERTEX Project Explorer with File ID 5 and Folder ID 2 as Video Content for this Console Layer

- you have to set DMX-channel 49 of your lighting desk to value 2 and DMX channel 50 to value 5
- The content item is selected as video content for the VERTEX Console Layer

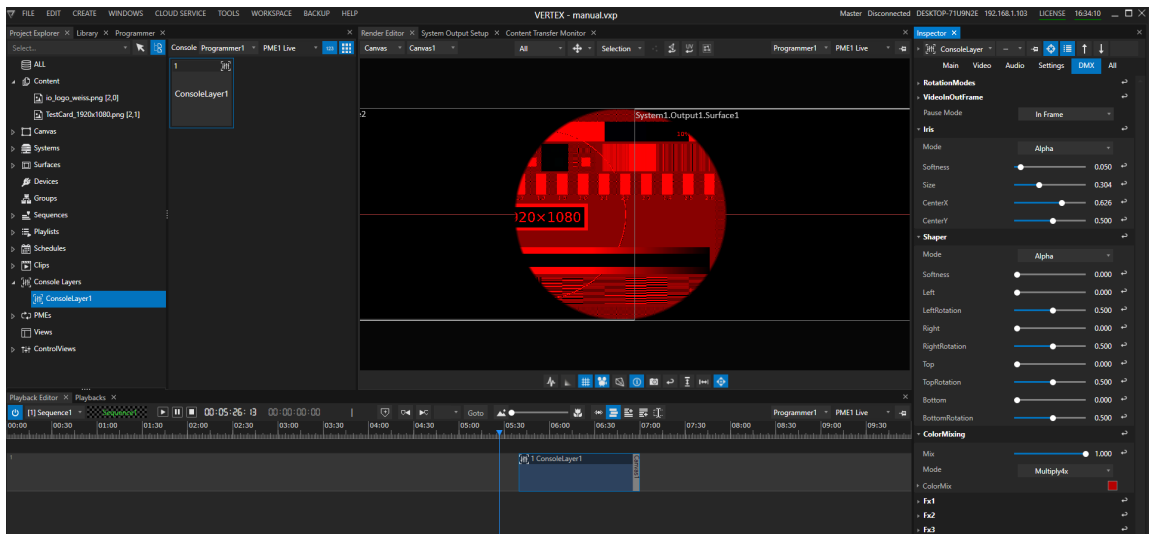
### Set File and Folder ID



- Select a Content item into Project Explorer and go to the Inspector
- Go to the Tab "Settings" and Select "DMX ID"
- Set File and Folder ID for a Content. Each value has to be into the DMX parameter range from 0 to 255

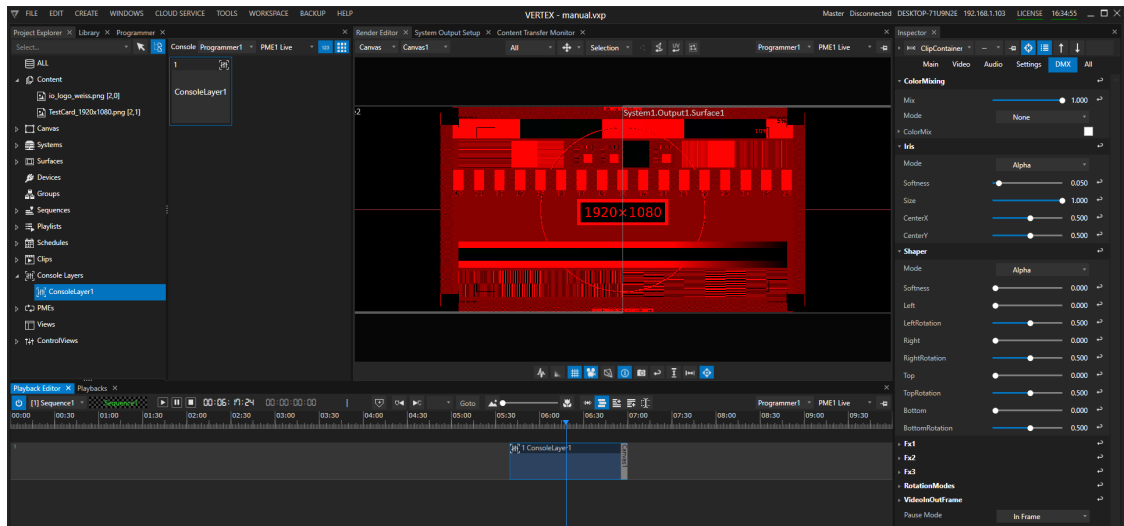
### Clip Container for a Console Layer

- Even for Console Layers you can create a Clip Container
- Used in a Playback, Clip Containers for a Console Layer overwrite the global value of the Console Layer



Global Setting for ConsoleLayer 1: Iris is set

- Drag a Console Layer from Project Explorer into the Playback Editor: A new Clip Container is created into your Sequence
- Assign Content to your Clip Container and set Values



Clip Container with Console Layer 1 inside overwrites the global values of Console Layer 1: in this example, all values for the iris are reset to default values.

Don't forget to also assign the same Main Content as the global Console Layer has to the Clip Container



**Use case example:**

You are able to e.g. temporary take control back from a lighting desk control and to set all content to black.

## 6.5 Control View

- ControlViews are **graphical touch interfaces** with e.g. buttons and sliders
- Use the **ControlView Editor** to **easily design** and build your own interface without any programming skills
- ControlViews can be **displayed and run in a ControlView Editor** or in **fullscreen mode** on a VERTEX System

### Basics

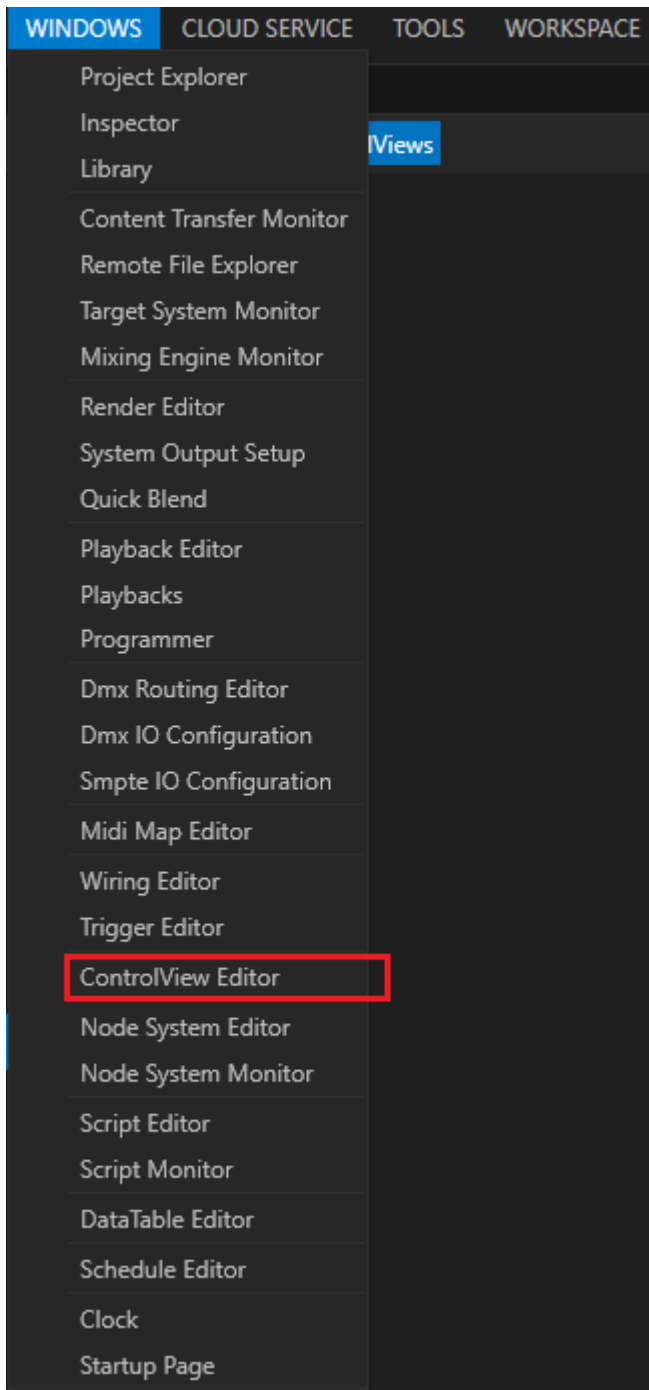
- ControlViews are **easy to made touch interfaces to control** your playbacks and devices.
- VERTEX is shipped with a bunch of **prestyled control elements like buttons and sliders**.
- The **ControlView Editor** helps you to visual build **touch interfaces** without layout and programming skills.
- Detailed **layout parameters** allow you to **adapt User Interfaces** to e.g. a client's cooperate design
- Control Views could be run into a **Control View Editor** or in **fullscreen mode**. For fullscreen, every VERTEX system provides a **Control Viewer**
- Fullscreen touch interfaces could also be used **as transparent overlay on a desktop** or even **on top of a Fullscreen Renderer**
- *iversal* also provides a **reduced VERTEX version** only for touch interfaces - called **VERTEX Touch**
- **Every control element** could be accessed by **a script command** or by the **API**

### ControlView Editor

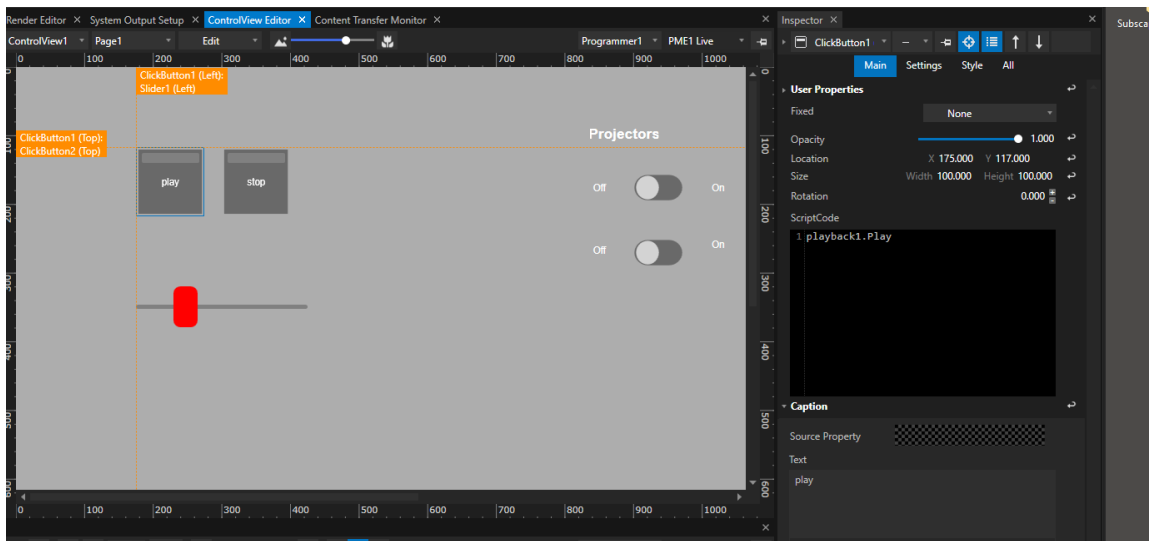
- The **ControlView Editor** helps you to design visual touch interfaces
- The Editor can run in **two different modes: Edit** (for interface design) and **Run** (for Interface use)

#### Open a new ControlView Editor

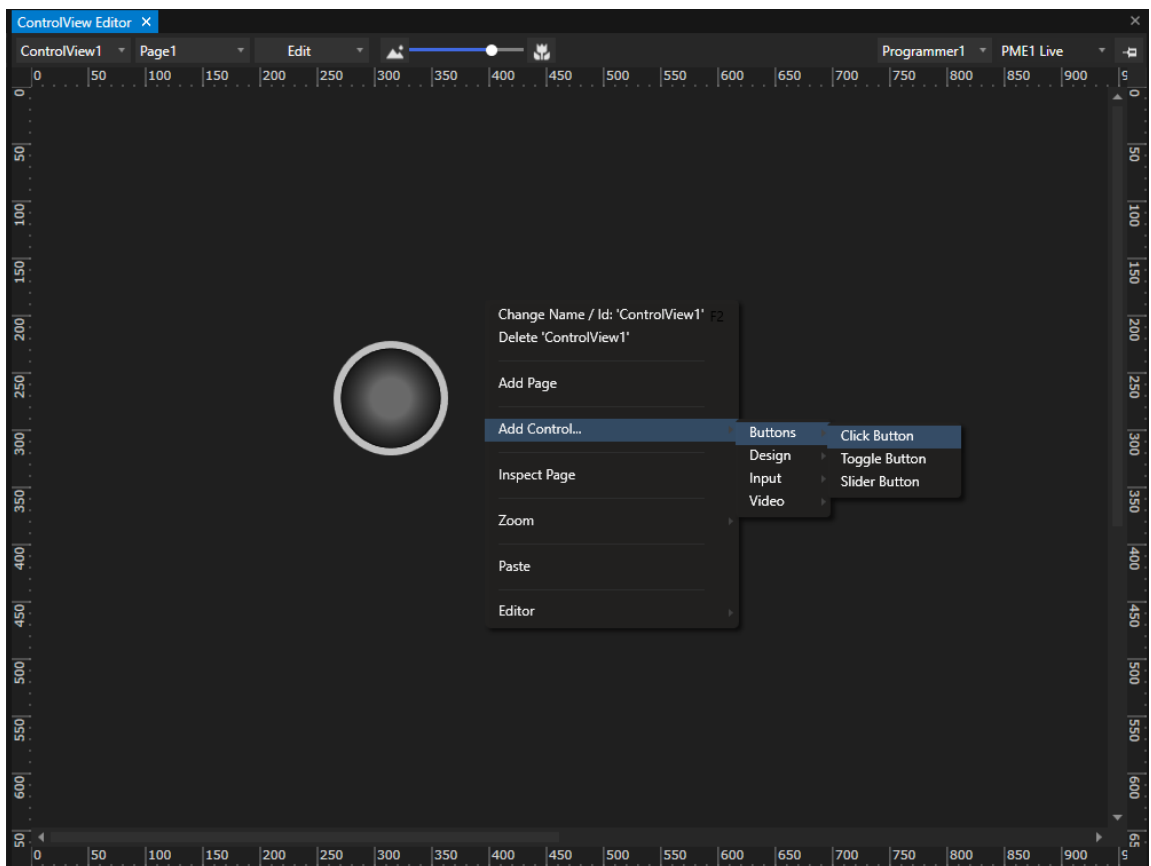
1. Go to the **"Create" Tab into the Main Menu** and **create a Control View**
2. Go to the **"Windows" tab into Main Menu** and open a new ControlView Editor



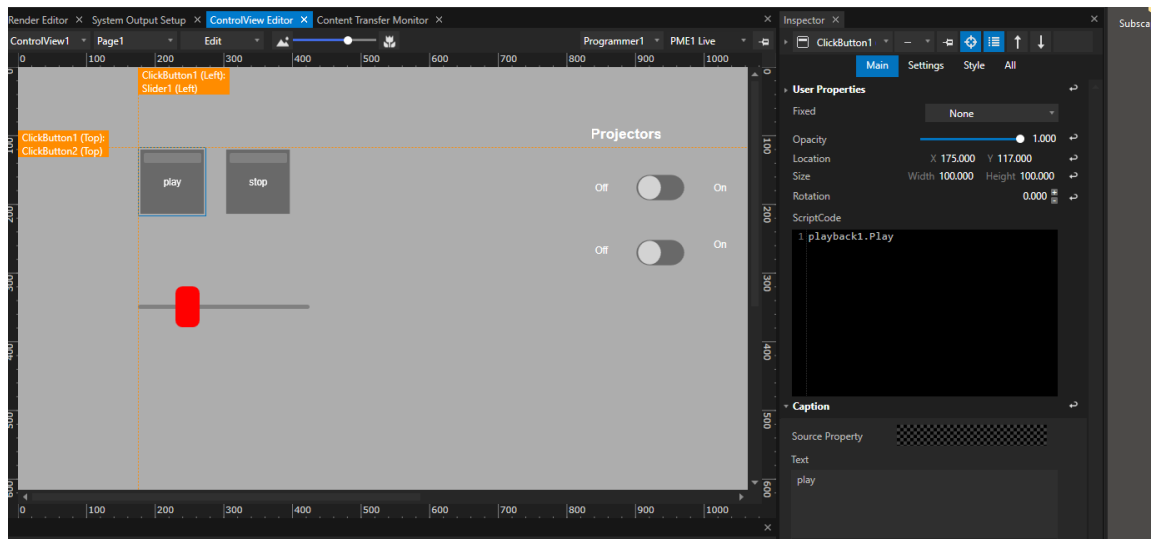
## Edit



3. Select a Control View you want to edit in the dropdown of the Control View Editor
4. Switch Control View Editor from "Run" to "Edit" mode
5. Right-click with your mouse , open the context menu and go to "Add Control" there



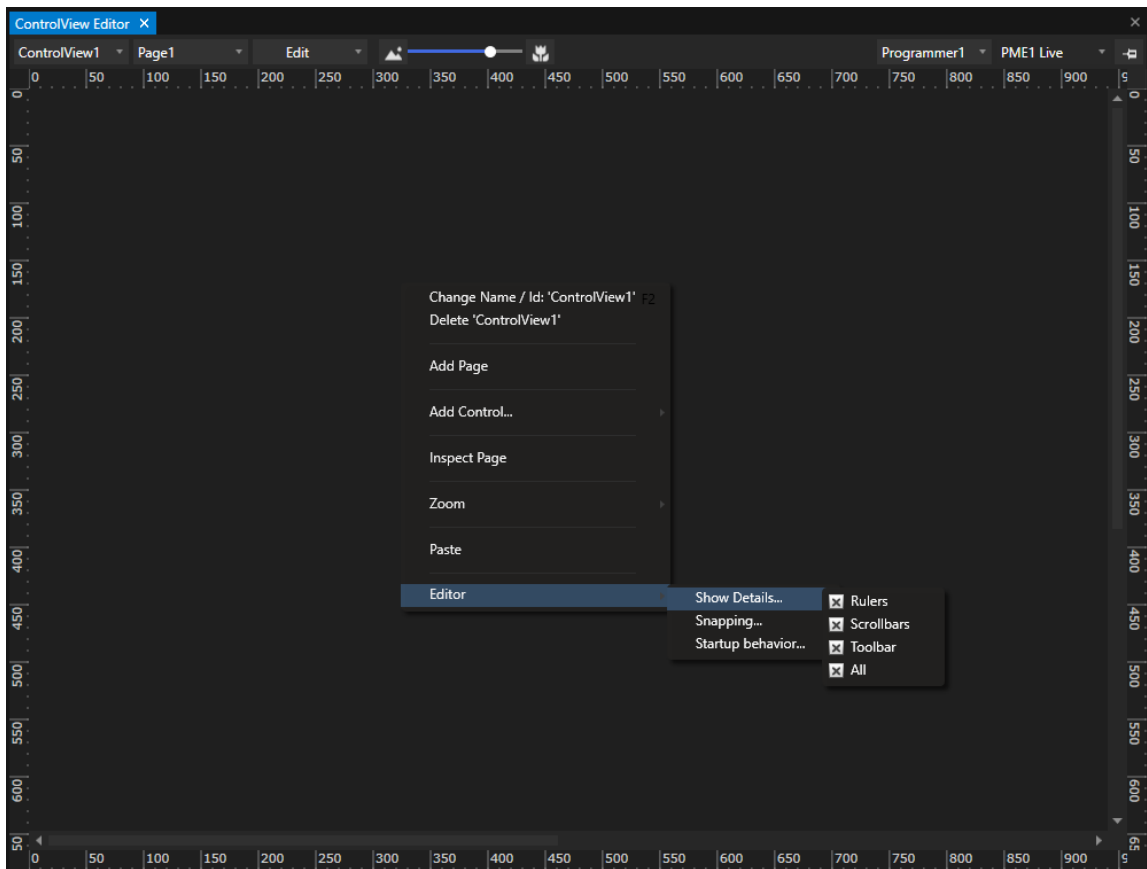
6. Choose between different control elements like buttons, sliders, label
7. **Move and Place** the button **with your mouse** into the ControlView Editor
8. By default, layout helpers , snapping and grid and rulers will help you to easily arrange your controls
9. Select a control and do **settings for interaction or style adjustments** into the Inspector



10. When you are finished with your layout and settings, **switch mode back from "Edit" to "Run"**
11. Or just start a Control Viewer in Fullscreen

## Editor Settings

- Open the context menu with a right click in edit mode
- select **"Editor"**
- there you are able to define the ControlView Editors layout helpers, set the snapping or even the startup behaviour

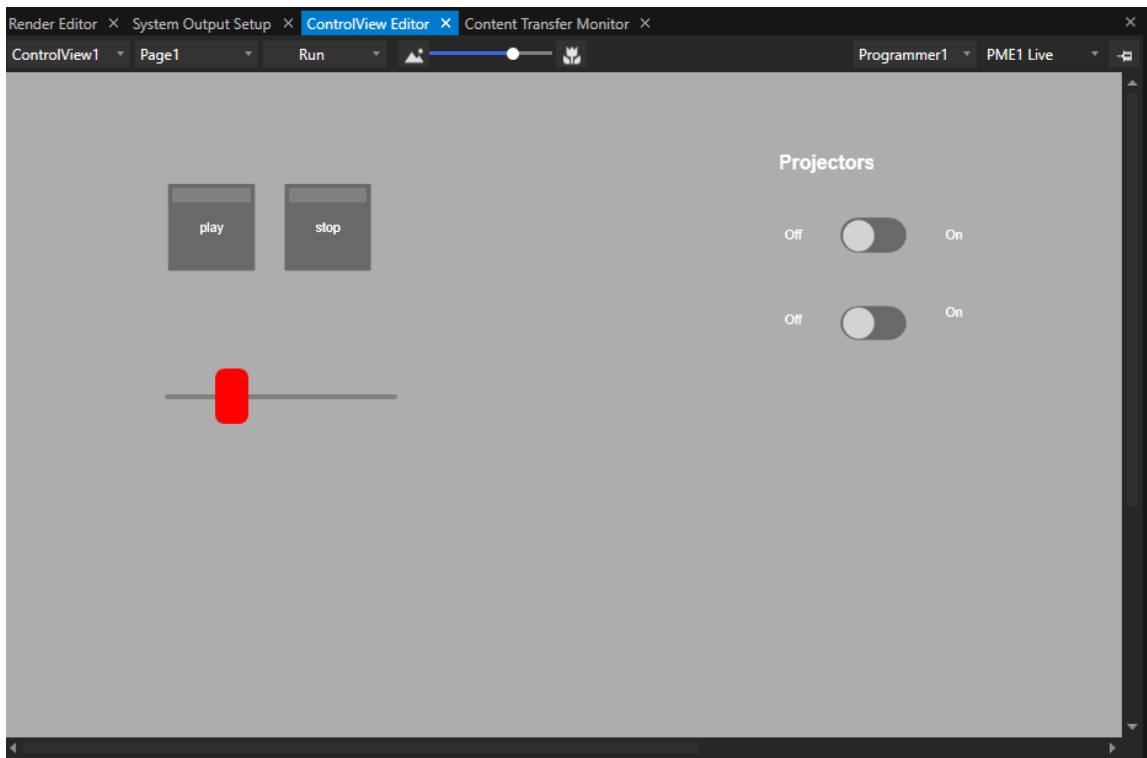


## Pages

- For every ControlView, you are able to create multiple pages as child elements of this Control View
- This pages allow you to create some submenus or to e.g. thematically order or purify your touch interfaces
- RightClick into ControlView Editor ( In edit mode), open the context menu and select "Add page"

## Run





When switched into **run mode**:

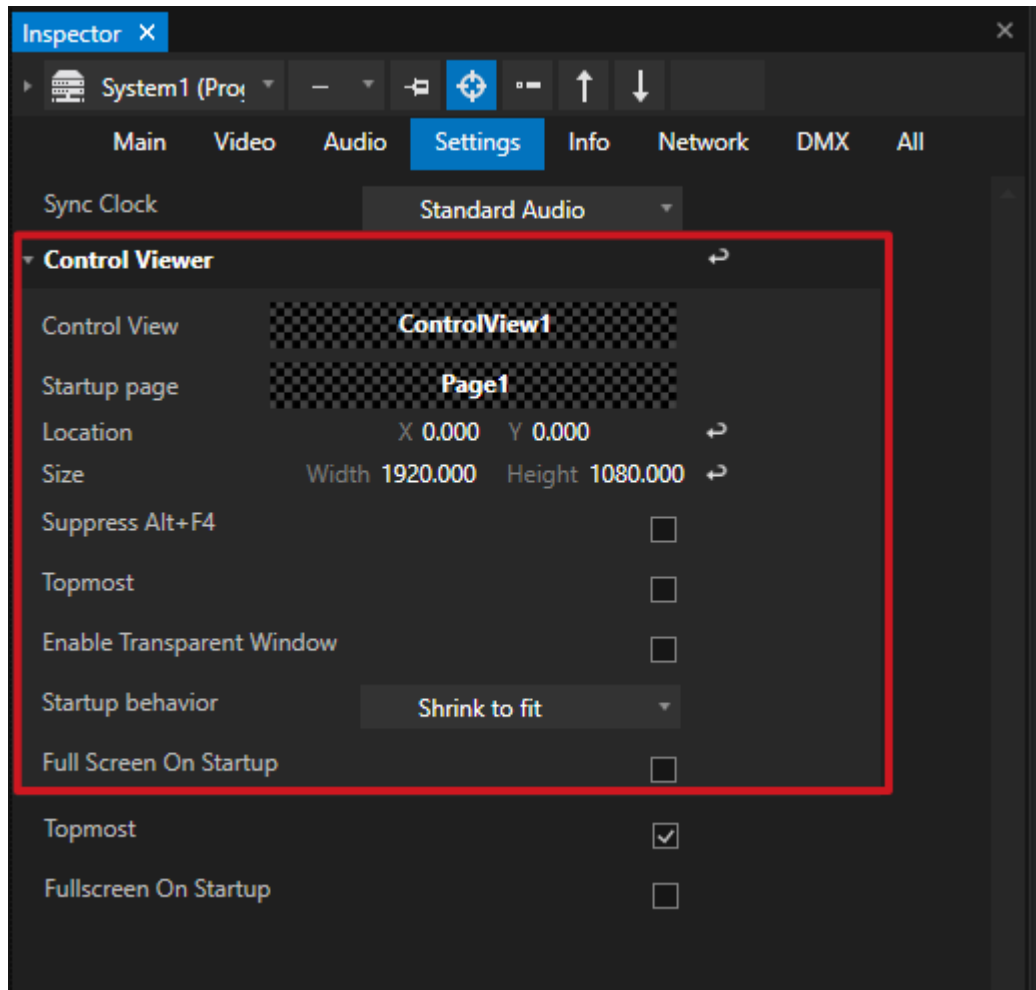
- all objects are locked
- all rulers and context menus are deactivated
- Objects could not be selected into inspector

## ControlViewer

- Each VERTEX System has a capability to show a **ControlView as full screen** - This capability is called **Control Viewer**
- The Control Viewer is located on **System level**. Select a System into Inspector and do the main settings for full screen there.
- Depending on your [VERTEX license features](#), it is possible to **run a Control View and a Video Renderer in full screen** on the same VERTEX System
- It is also possible to **overlay a transparent ControlView** to a full screen Renderer or your Windows Desktop

## Settings

- **Select the System** on which your Control View should be shown as full screen
- Go to the **Control Viewer** section in the Inspector



- **Size** defines the pixel size of the full screen Control View
- **Location** defines the position of this full screen Control View - the top-left start coordinates of your desktop

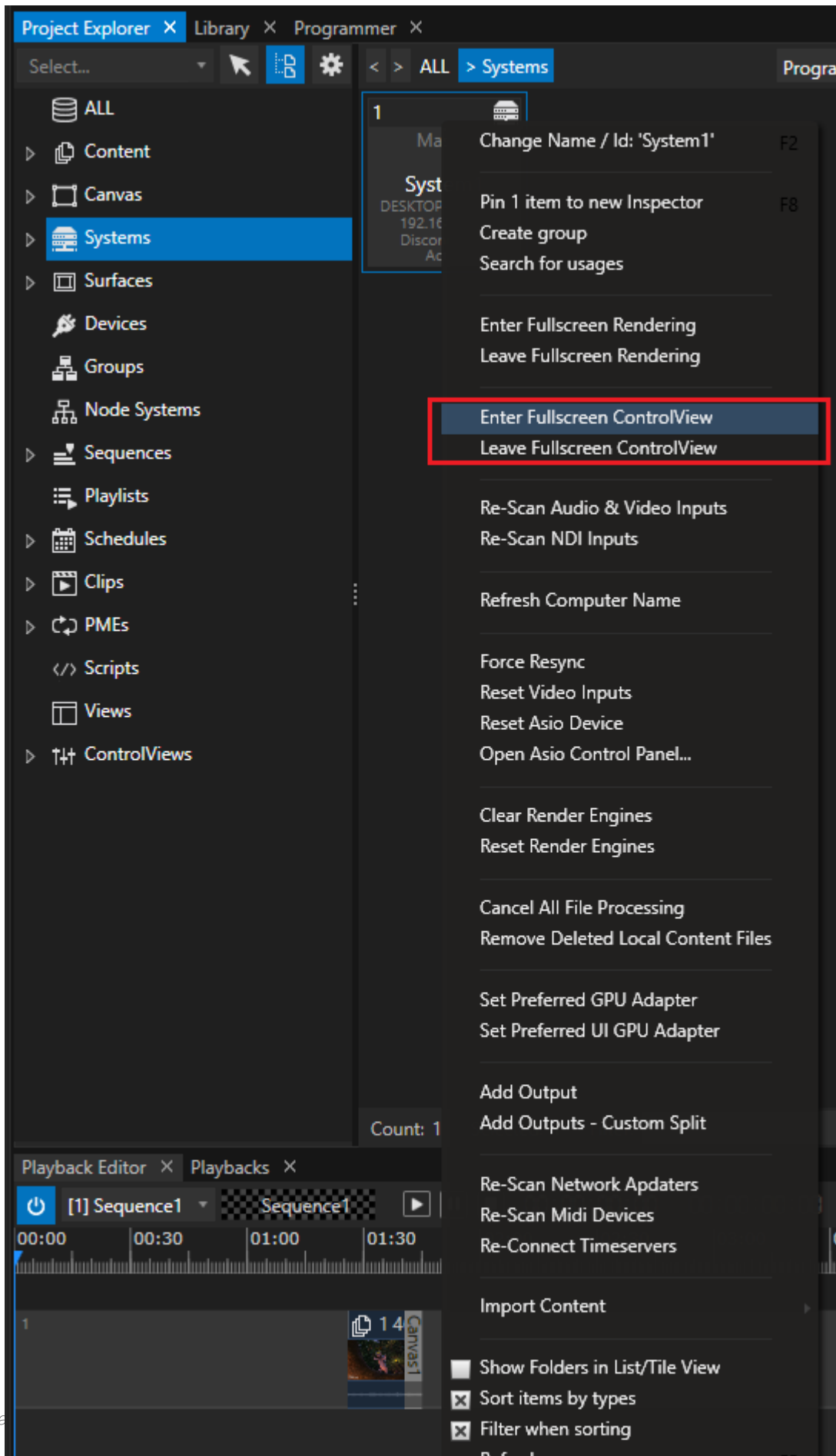
## Set to Full Screen

Choose between **4 options** to set the Control Viewer in Full Screen:

### 1. Context Menu into Project Explorer

Right-click on a system and select "Enter Fullscreen ControlView"

If working with multiple Systems in a Project, you can remote set another system into Full Screen



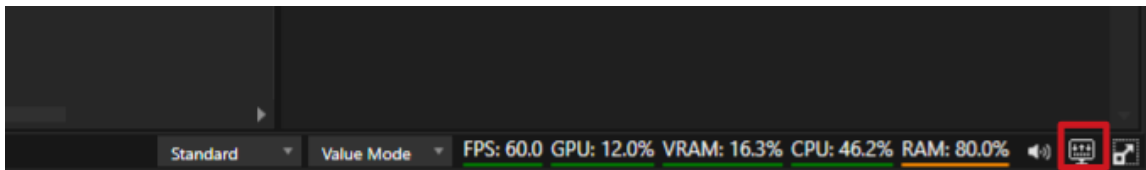
## 2. Shortcut (CTRL + F5) on your local System

Press the same shortcut on your local System to leave again.

## 3. Control Viewer Button in Status Bar

Use the Control Viewer Button into [Status Bar](#) to switch your local system into start a Control Viewer on your local System in Full Screen

Use Shortcut CTRL+ F5 to close Control Viewer again



## 4. With a Script Command

Of course there are [Script Commands](#) to open and close the Control Viewer

```
System1.ControlViewer.Open  
System1.ControlViewer.Close
```

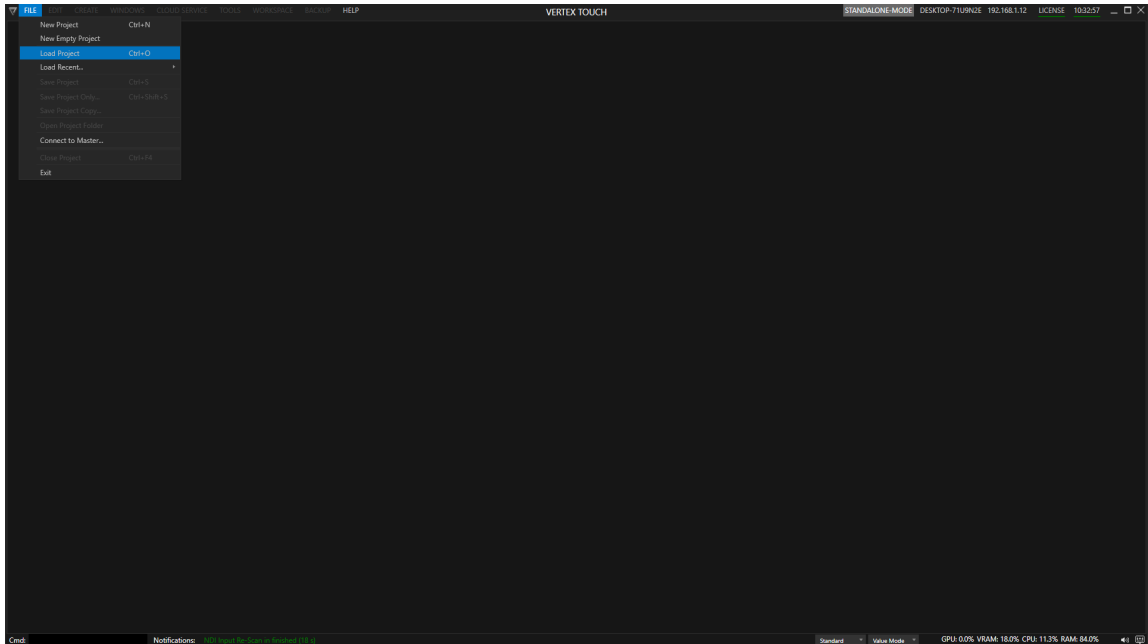
or for your local system just

```
ControlViewer.Open  
ControlViewer.Close
```

# VERTEX TOUCH

- VERTEX Touch is a VERTEX software version with a **reduced functionality and reduced user interface** - optimized for fullscreen touch interfaces
- The application starts **with a smaller UI footprint** without a Render Editor and no option for a Fullscreen Renderer - it is **optimized even for less powerfull PCs**

- VERTEX Touch is **shipped with every regular VERTEX Installer** - a shortcut is created as Windows application during the regular VERTEX installation process.



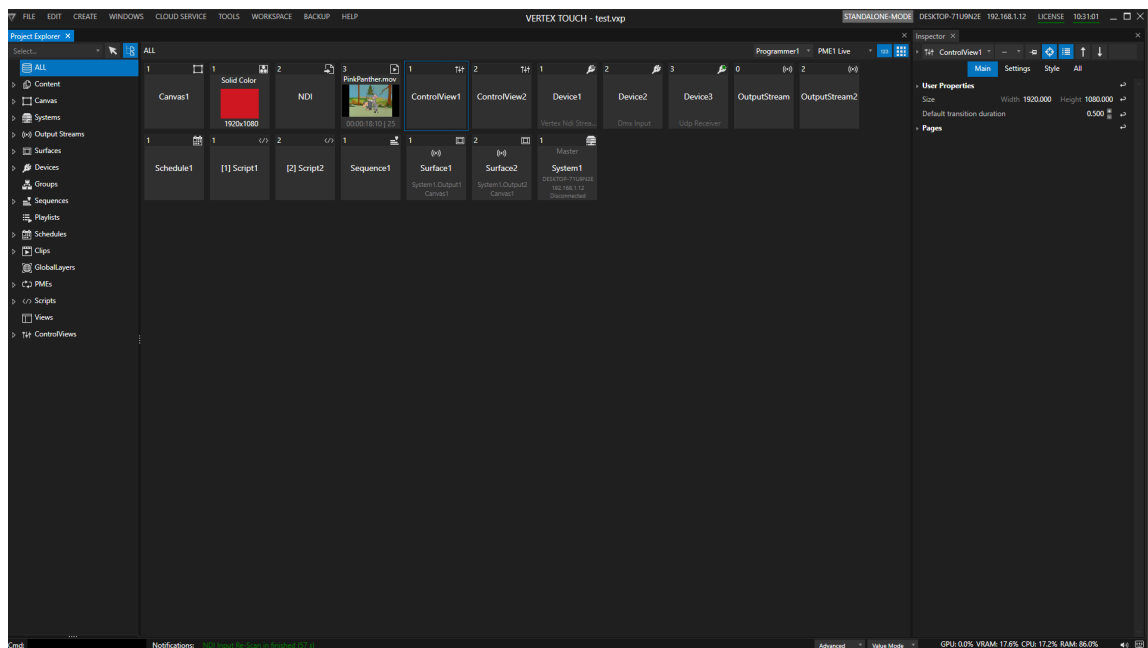
*User Interface after VERTEX Touch application was started.  
Compared to the regular VERTEX application there is no startup page*



### VERTEX Touch License Edition

ioversal also offers a [license edition](#) called VERTEX Touch for a reduces price.

You are able to run the VERTEX Touch application with a license of every other [VERTEX edition](#).



User Interface VERTEX Touch

## Access a ControlView with a Script

Like most things into VERTEX also ControlViews are accessible with [a Script Command](#).

Please note, that there is a **difference between a ControlView** that contains of Labels, Buttons, Sliders, Pages and its settings

**and the ControlViewer**, that hosts the ControlView

## Difference between ControlView and ControlViewer

To call layout elements ad settings of a Control View, start your Script with the VERTEX Object ControlView ( e.g. "ControlView1")

```
ControlView1.Pages.Page1.Label1.Delete
```

To switch to Page 2 of ControlView1 use start your Script with the ControlViewer object e.g.

```
ControlViewer.GotoPage Page2
```

## Script examples

Change the label text of Label 1 in ControlView1

```
ControlView1.Controls.Label1.Settings.Caption.Text.Value = "this is a
```

Perform a Click on Button 1 of ControlView1

```
ControlView1.Controls.ClickButton1.Click
```

Return the current Page that is displayed by ControlViewer1

```
ControlViewer.GetPage
```

Switch to Page 2 of the current control View. Also works as script for e.g. button in ControlView Editor (Run Mode)

```
ControlViewer.GotoPage Page2
```

Delete Label 1 on Page 1 of ControlView1

```
ControlView1.Pages.Page1.Label1.Delete
```

## 6.6 Devices

- Control **different types of external devices** directly from VERTEX
- Select a device from **library** and add it to your project
- Set device parameters with **script commands**

### Workflow

VERTEX comes up with an **abstract device model** that enables you to **control different types of devices with various functionality**.

Some of the device properties are animatable via keyframes like the ones of DMX-Devices for instance. But basically all device parameters are accessible with VERTEX script commands.

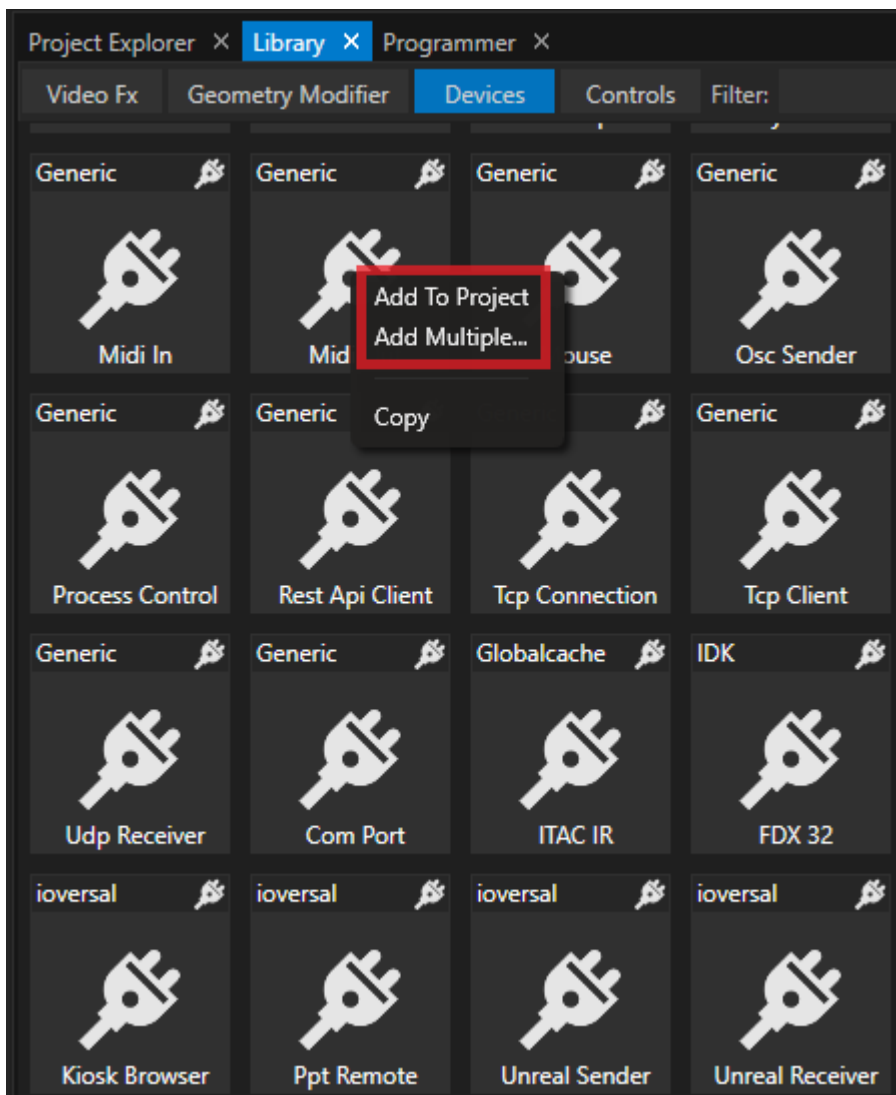
**Regardless of the device type and its range of parameters, the basic workflow is always the same:**

1. [Open the library editor and search for a device](#)
  2. [Add the device to your project](#)
  3. [Select device in the project explorer](#)
  4. [Do connectivity and protocol settings in Inspector](#)
  5. [Access device parameters with vertex script commands](#)
- or
6. [use a device in a clip container](#)

[Troubleshooting: Reconnect a Device](#)

### Devices in Library Editor





- Open the *library*.
- Search for the device you want to use  
use the search filter and search by names
- Right-click with your mouse and select whether you want to add only one or multiple devices to your project

**Directory and Folder for Devices**

You are able to write your own device templates and add them to the VERTEX library.

You have to save your custom devices with the file extension `.vxd`

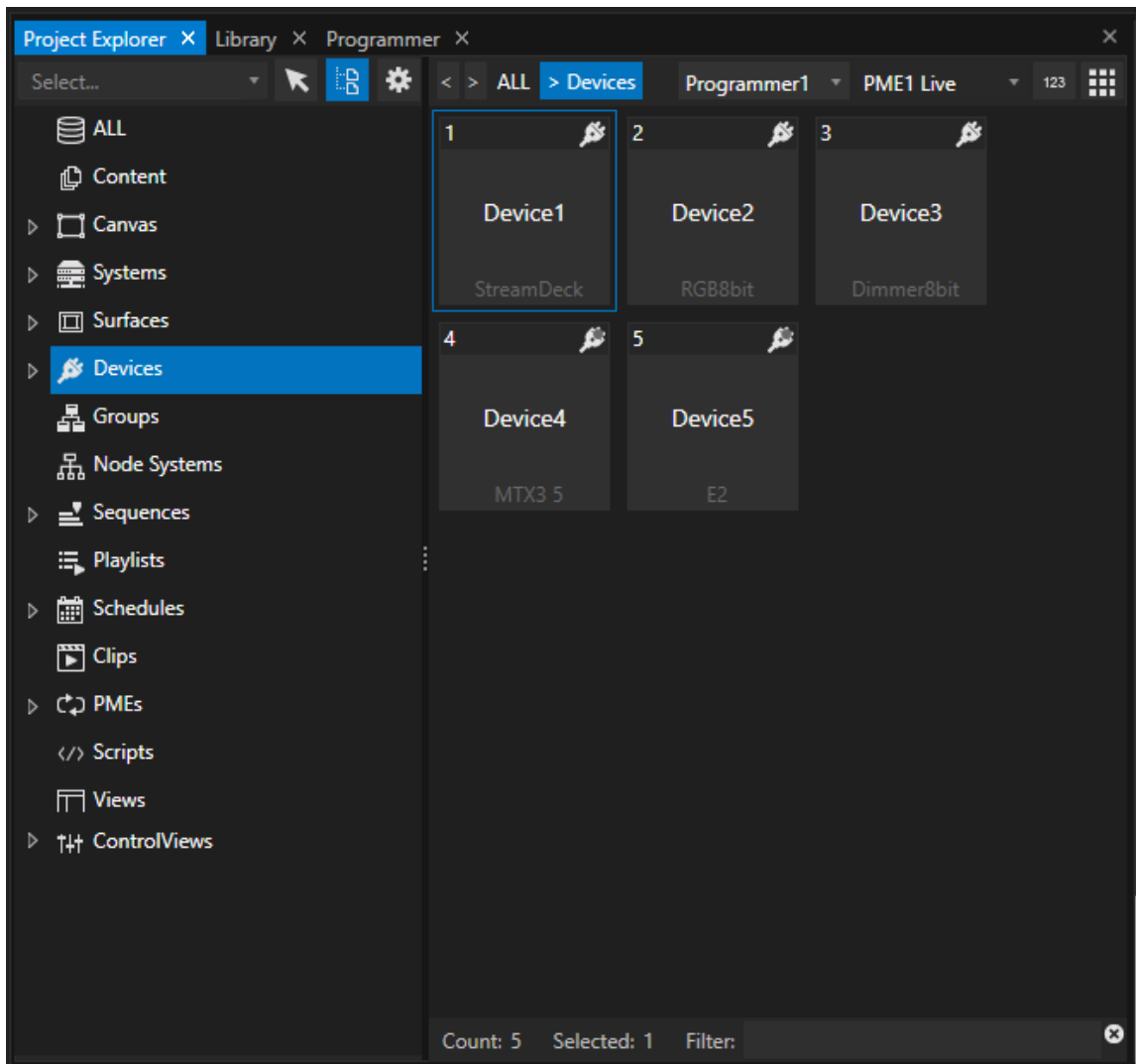
Devices that are shipped by ioversal are encrypted `.vxdx` format

**Directory and Folder:**

`C:\Users\Public\Documents\ioversal\Vertex\[Vertex Assembly Version]\Devices`

You are allowed to create own subfolders or copy your device into an already existing subfolder

## Devices in Project Explorer



The **device section in the project explorer** contains all devices that you have added to your project. Depending on the device type, a **connection status** gives you information about the connectivity.

**Status:**

**Red:** connection refused or not possible

**Orange:** try to establish connection

**Green:** connected

**Connection settings** like ports or IP Address could be done in the **inspector**.

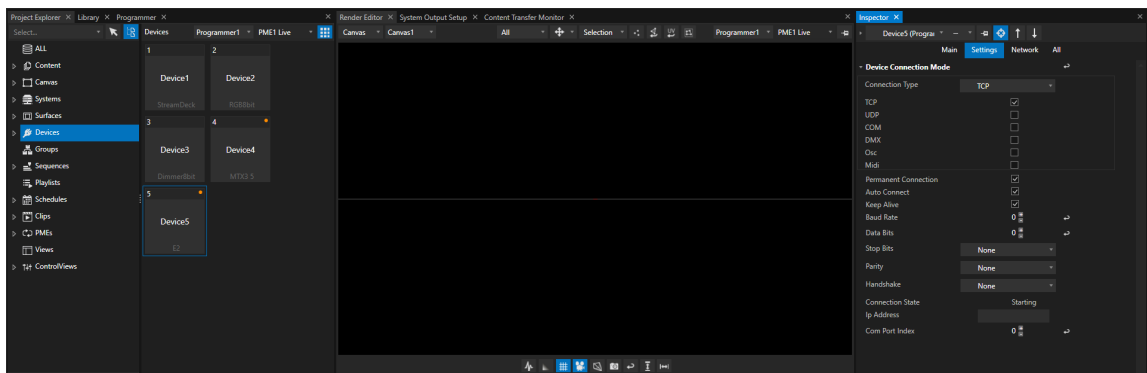


### Connection status into Project Explorer only for certain types of Devices

Some device types like DMX Devices or TCP Senders do not have a connection status by device design

For this Devices the status is not shown into Project Explorer

## Connectivity and Protocol Settings

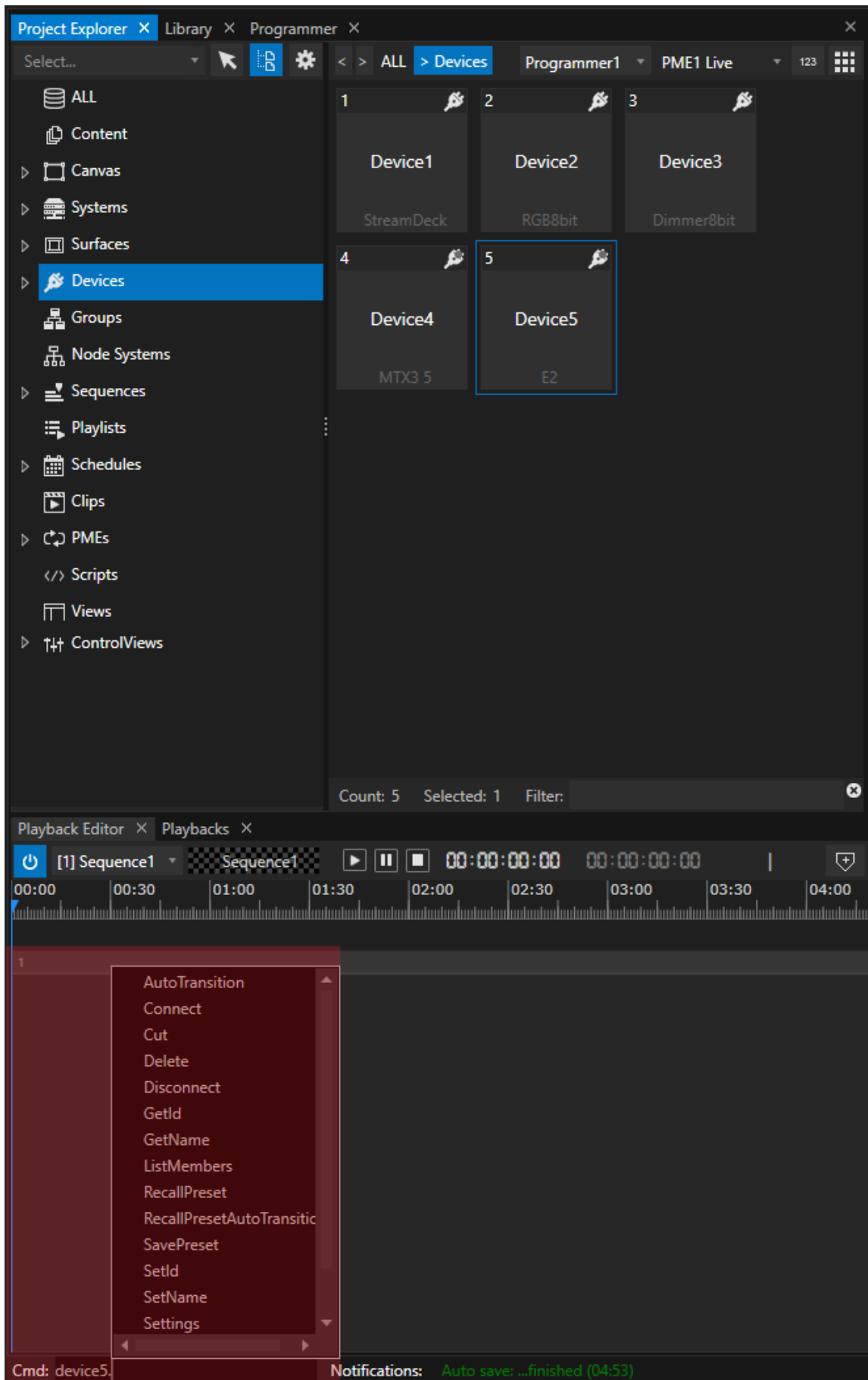


Select a Device in the project explorer and do the **connectivity and protocol settings** in the inspector.

Within the device library that is shipped with VERTEX, ioversal has tested the connectivity for most of the included devices. **All connectivity and protocol settings are set according to the manufacturer's specification.**

If you have problems to connect one of the devices, don't hesitate to contact the ioversal support ([support@ioversal.com](mailto:support@ioversal.com)) or open a ticket directly from within VERTEX (Main Menu -> help -> Create new Ticket).

## Use Devices with Script Commands



*Each device from a manufacturer has its own parameters and options.*

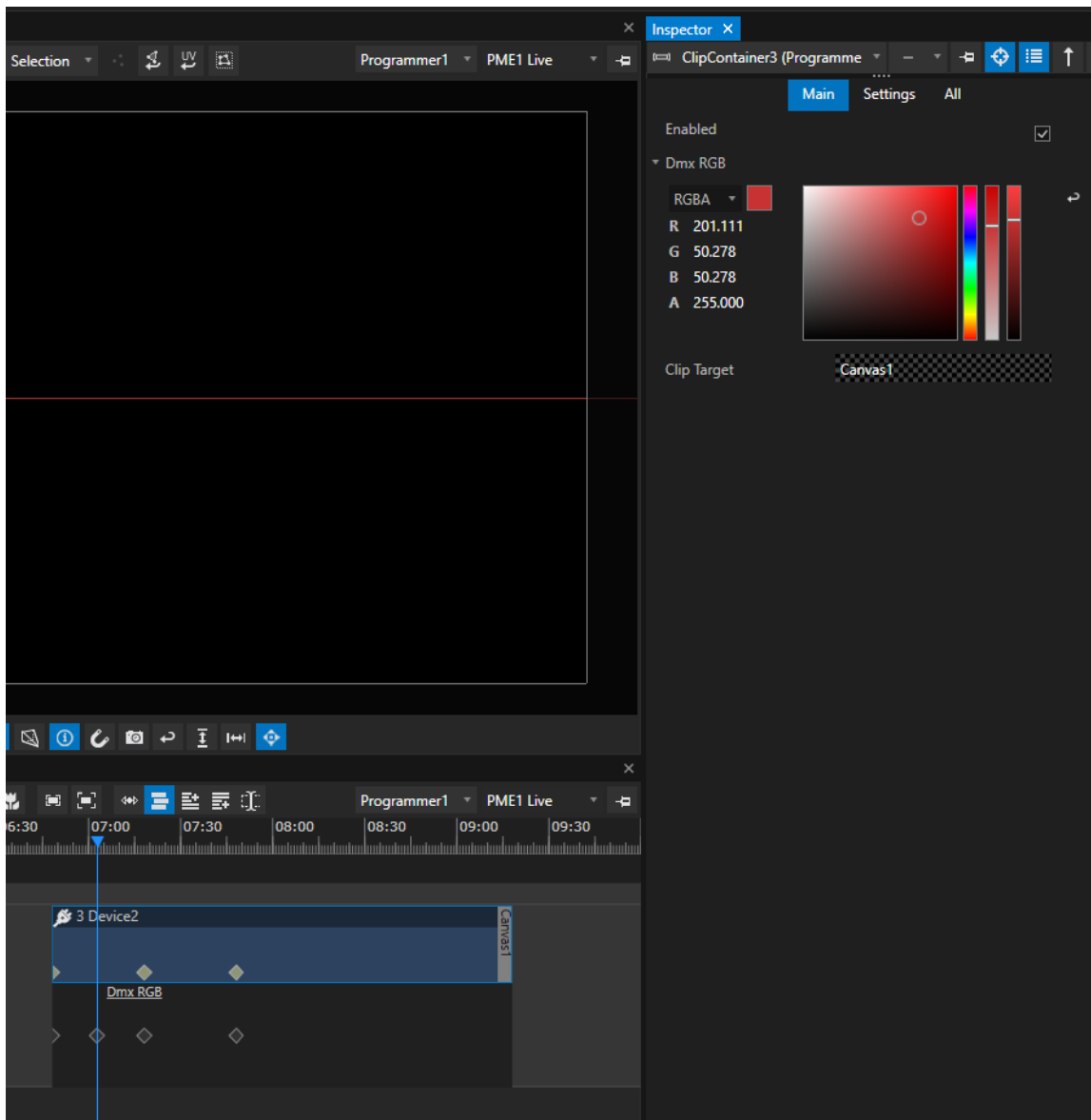
*This **options and parameters** are **accessible with VERTEX Script commands**.*

*Use script commands in cues, clip containers, on system startup, in a playlist, or in a trigger editor to script your device interaction.*

***For some kind of devices** - like DMX Devices also **keyframe animation** is possible.*

*For **some devices** (like e.g. the Elgato Stream Deck) **you only have to do the global settings into Inspector to work with.***

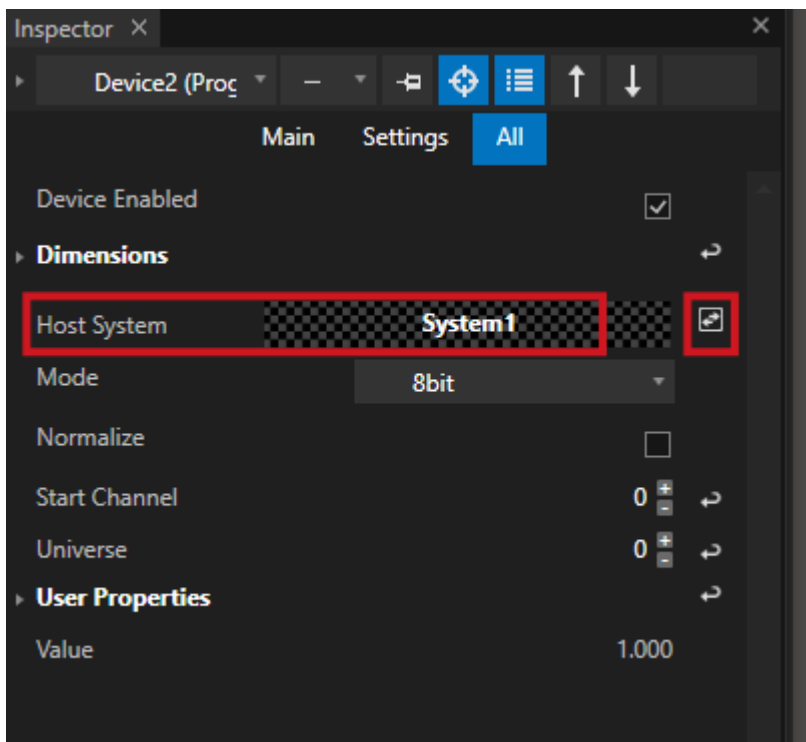
## Use Device as a Clip Container



Some Devices like **DMX Dimmer** and **DMR RGB** devices could be **animated with keyframes**.  
Just **drag the device from project explorer with your mouse and drop it into the playback editor**.  
A **clip container** is created.

## Connection Problems - Host System and Reconnect

- If there are **connection problems with your device**, please try a **reconnect** first of all.



- Select your device in the *inspector*
- Every Device has a **property for Host System** (The VERTEX system your device is connected to). Please first check if this is correctly set  
If not and if you are working with **multiple systems**, drag another system from project explorer to this property field.
- Next to this property, there is a **reconnect button** that refreshes to connection to this device.

## Reconnect by Script

- A reconnect also is possible by script with the command **ResetConnection**

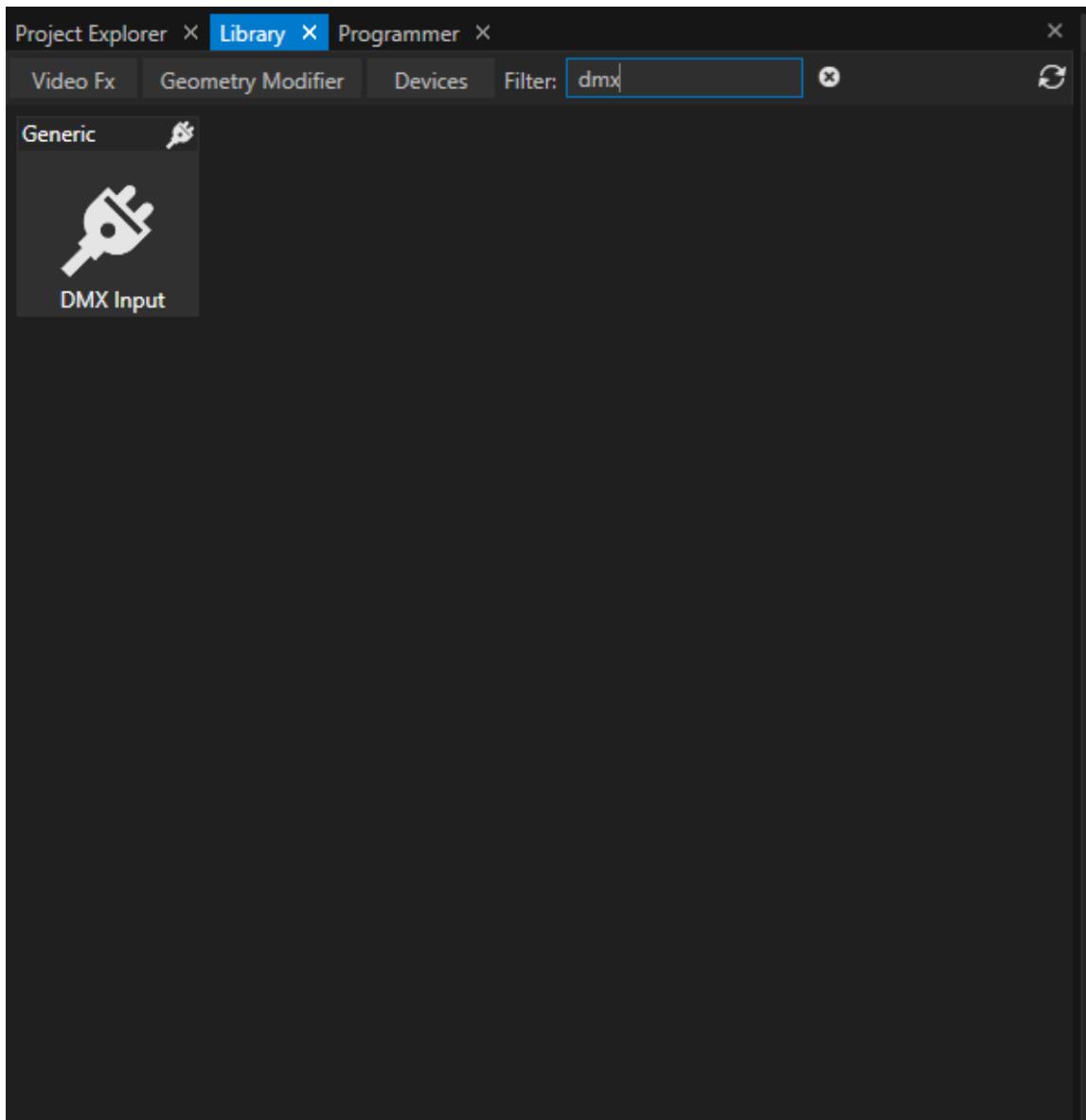
```
Device1.ResetConnection
```

## 6.6.1 DMX Input Device

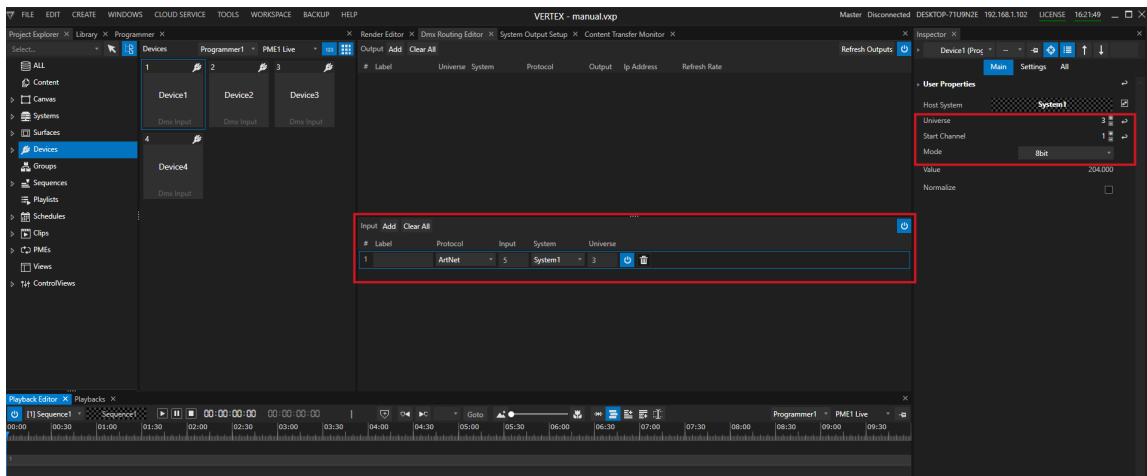
- Use DMX Input Devices to **trigger, wire or script interactions based on values of a DMX Channel**
- You are able to **listen on DMX Values** of a predefined DMX Start Address and Universe
- Use the **incoming DMX Values to influence other properties into VERTEX** or to trigger an action

## Add Device and do Settings





- Go to the Library Editor
- Select the tab "Devices"
- Search for "DMX Input"
- Right-Click on the Device and add one or multiple to your project



Example of a DMX Device Setting: There is DMX Input Routing. VERTEX is listening on Art-Net Universe 5. Data from this Universe is internally routed to VERTEX Universe 3. DMX Input Device 1 is set to VERTEX Universe 3, Start Channel 1.

- Go to the Device section in the [Project Explorer](#)
- Select on or multiple of your DMX Input Devices
- Set VERTEX universe number and start channel and define the DMX mode (Default 8 Bit, 16 Bit, 32 Bit...)



### Normalize Input

Enable "Normalize" in the Device settings, if you want to control or wire a property into VERTEX that has a value range from 0 to 1 (e.g. opacity).

All incoming DMX Data is normalized into the range from 0 to 1.

- Ensure, that a valid [DMX Input Routing](#) exists.

## Working with DMX Input Devices

As for every Device, there are different options to work with the incoming Data.

Here some Examples:

### 1. Scripting

The Opacity of Clip Container 1 is controlled by the incoming DMX Data from Device 1

The opacity value range is 0 to 1. The "Normalize" data option is enabled for the input Device

A [Script Command](#) for this scenario could be like this:

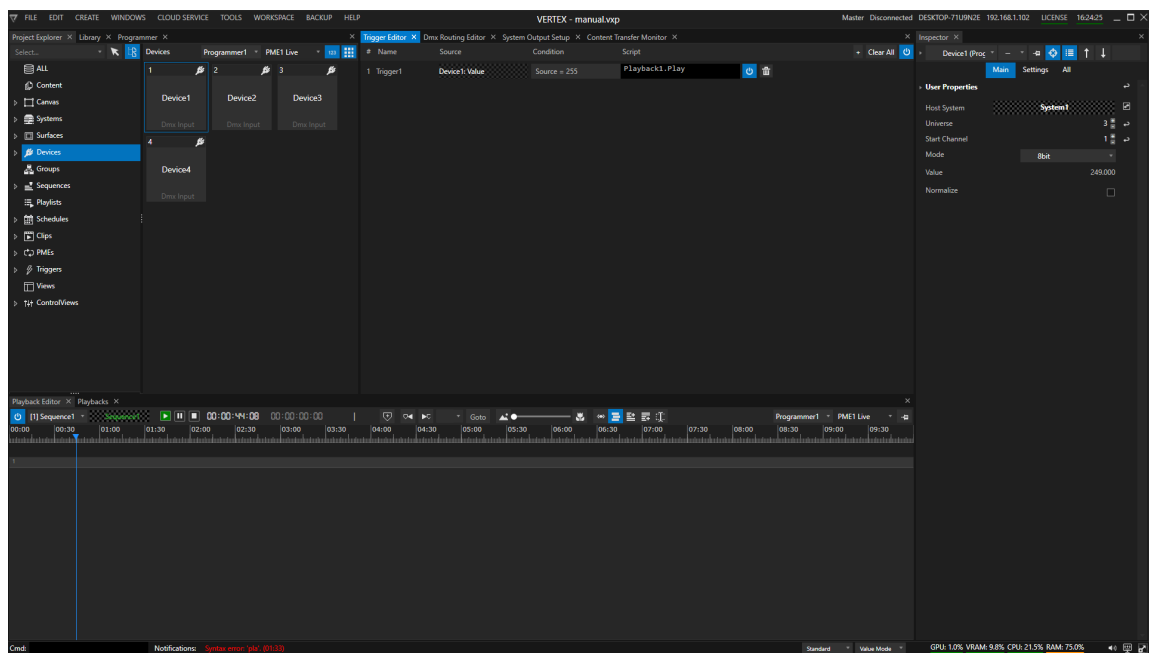
```
Sequence1.ClipContainer1.Opacity.Value = Device1.Settings.Value.Value
```

### 2. Trigger

With help of the [Trigger Editor](#) you can easily build Triggers. An defined action is executed if a condition for a source is reached

Of course this also works with DMX Input Devices

- Open the [Trigger Editor](#)
- Drag the e.g. Value property from Inspector to the Source field
- Enter a [Condition](#)
- Enter a [Script](#) that should be executed



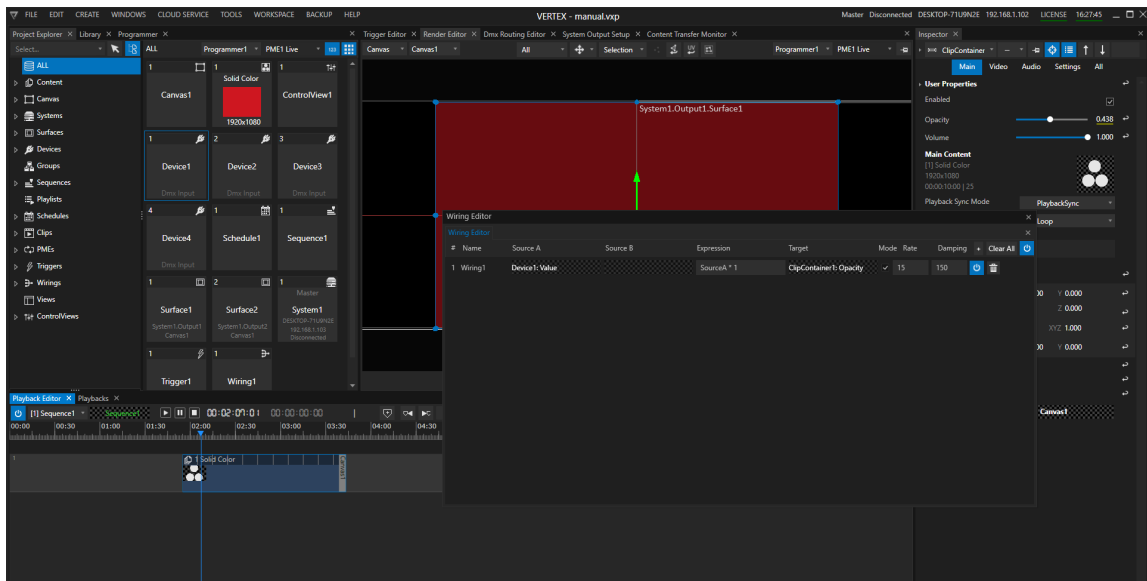
Example for the usage of a DMX Input Device:  
If DMX-Value of Universe 3/Startchannel 1 is 255, Playback 1 is started

### 3. Wiring

[Wiring](#) gives you the option to connect Values of different properties.

Of course this also works for DMX-Input Devices

- Open the [Wiring Editor](#)
- Drag a DMX Device Property from Inspector to e.g. the Source field
- Drag another property into target field
- Add an expression



Example of a Wiring

The Value of the DMX Input Device 1 ( Normalized) is wired to the Opacity of Clip Container 1.  
The Opacity now is controlled by a DMX Channel.

## 6.6.2 DMX Output Devices

- With *different types of DMX Output Devices* you are able to send out DMX Data
- Integrate DMX Devices to your show, *program lighting scenarios based on keyframes* or work with them in *programmer mode*
- Create a *group of Devices* and control them together with only one Clip Container

## Workflow

### Add from Library

- Open the **Library Editor**
- Select the **Devices** Filter there
- **Search for your DMX Device**



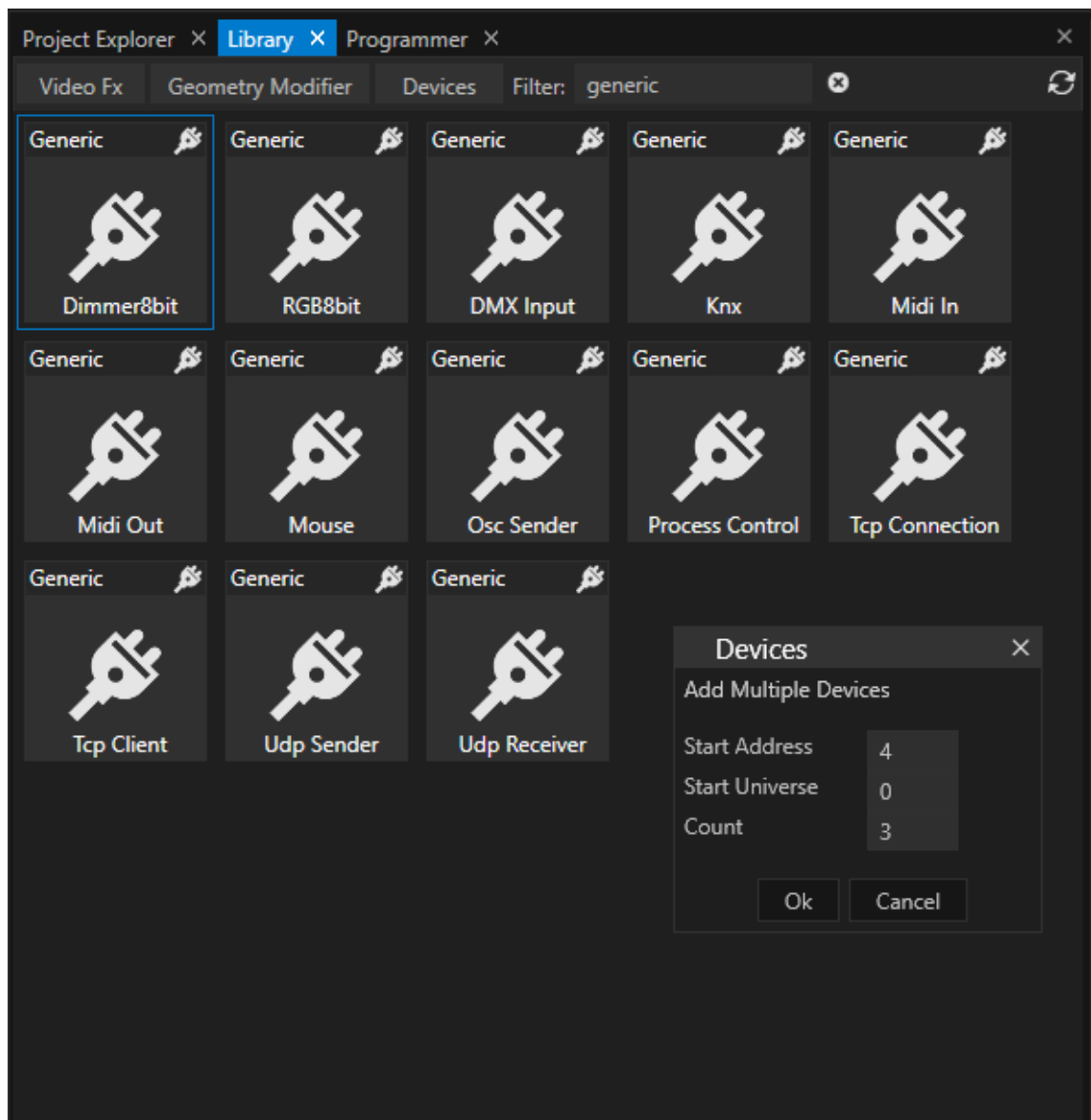
#### **DMX Devices and VERTEX Library**

Depending on the VERTEX Assembly version, the available number of different DMX Devices could change. Basically, VERTEX is shipped with Generic standard devices: a 8 Bit Dimmer and and 8 Bit RGB Device. If you require other DMX Devices for your project, just write us an email to [support@ioversal.com](mailto:support@ioversal.com) with type and in the most ideal case directly with the DMX

table :-)

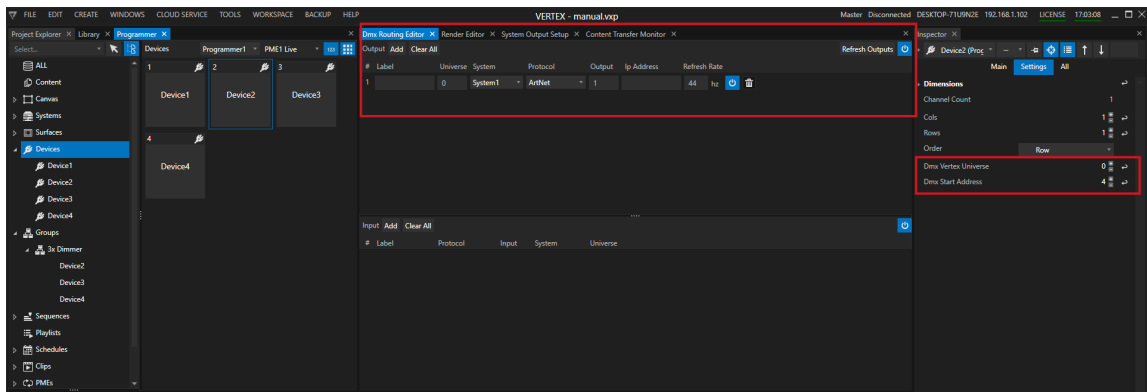
- **Right-Click** on the Device you want to add
- Select "**add to project**" or "**add multiple**"
- If you only have **added one device**: select the device into Inspector to set DMX Universe and Start address
- If you have select "**added Multiple**", a dialog window opens. Just add the **start address and universe** there and the number of devices you want to add.

The first device will start with start address and universe you have set. For all other devices, start address and universe are automatically set by VERTEX.



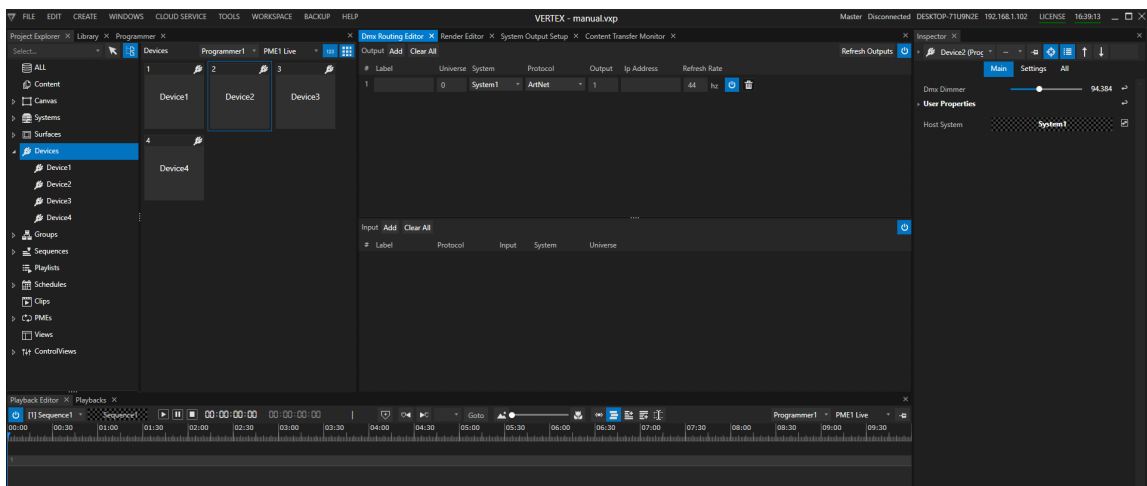
## Check DMX Routing and Start Addresses

- Remember that a DMX output only will work, when a **valid [DMX routing](#)** is created. If this was not done before, please create a routing now.  
Into the [DMX Routing Editor](#) you define, which protocol is used to send DMX Data out. To send physical DMX-512, ioversal provides an USB-DMX interface, the [<%DMX\\_IO%>](#)
- Go to the Device section of the Project Explorer. select a DMX device there and set or check the **DMX Start Address** or the **Start Universe** in the Inspector.

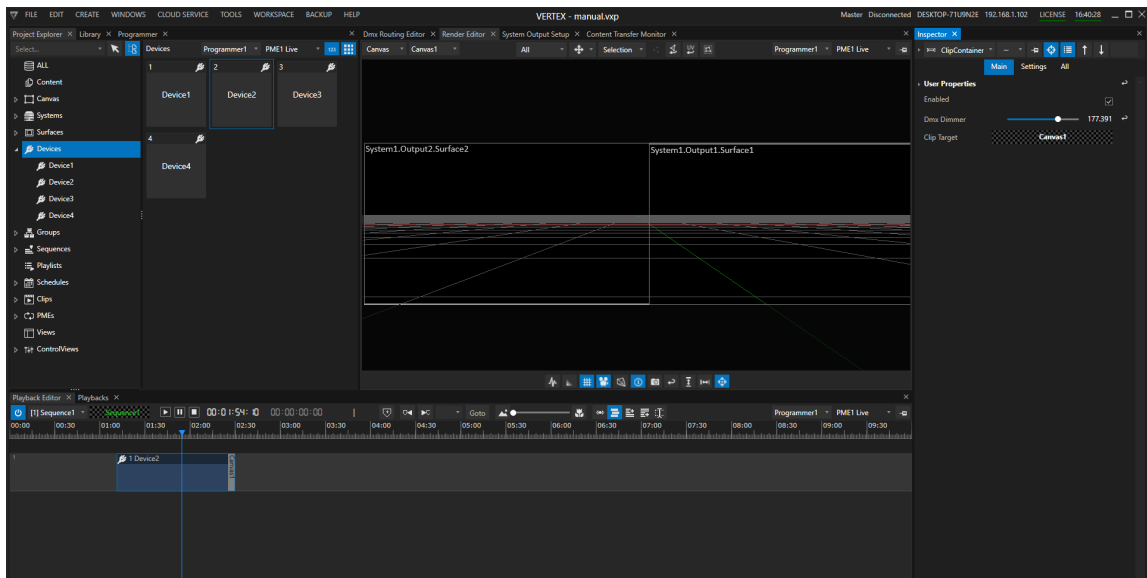


## Send Out Global Values

- You are able to send out a global value for your DMX device, e.g. for testing or if there generally should be a minimal value for such devices all the time.
- To set values globally, select a device directly into Project Explorer and set the DMX Value in the Inspector.
- This value is sent by vertex permanently

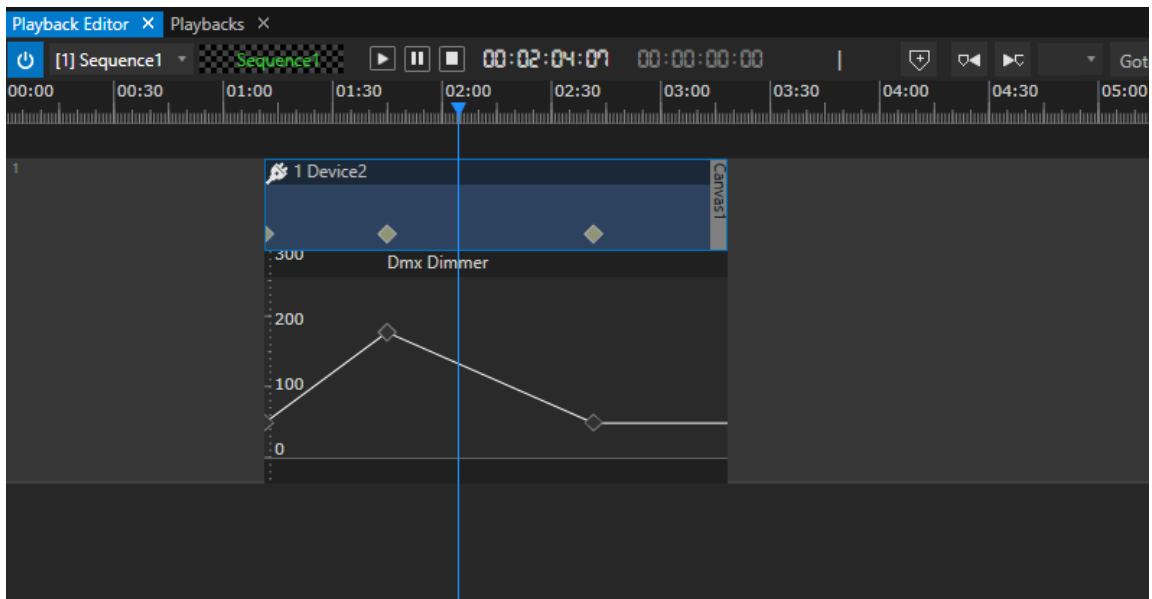


## Work with Clip Containers and Keyframes



- A combination of [Clip Containers](#) with [Keyframes](#) allows you to create more complex lighting scenarios, color transitions or an integration of lighting into your video show.
- Device values from a Clip Container temporarily **overwrites** [global values](#) (if such are set)
- Just drag a device from Project Explorer into the playback Editor to **create a new Clip Container**
- Double click on the Clip Container to open the **Keyframe Editor** (to learn more about Keyframe animation, please read the topic [Keyframes](#))





- If you prefer a **workflow from a Lighting Desk** - you can also work in **Programmer Mode**: Change all values and create your scene. All changes are temporarily stored as a list in the **Programmer**. When your scene is final, you can save the programmer list as a scene: VERTEX automatically creates the Keyframes for all changed values at the Playhead position.

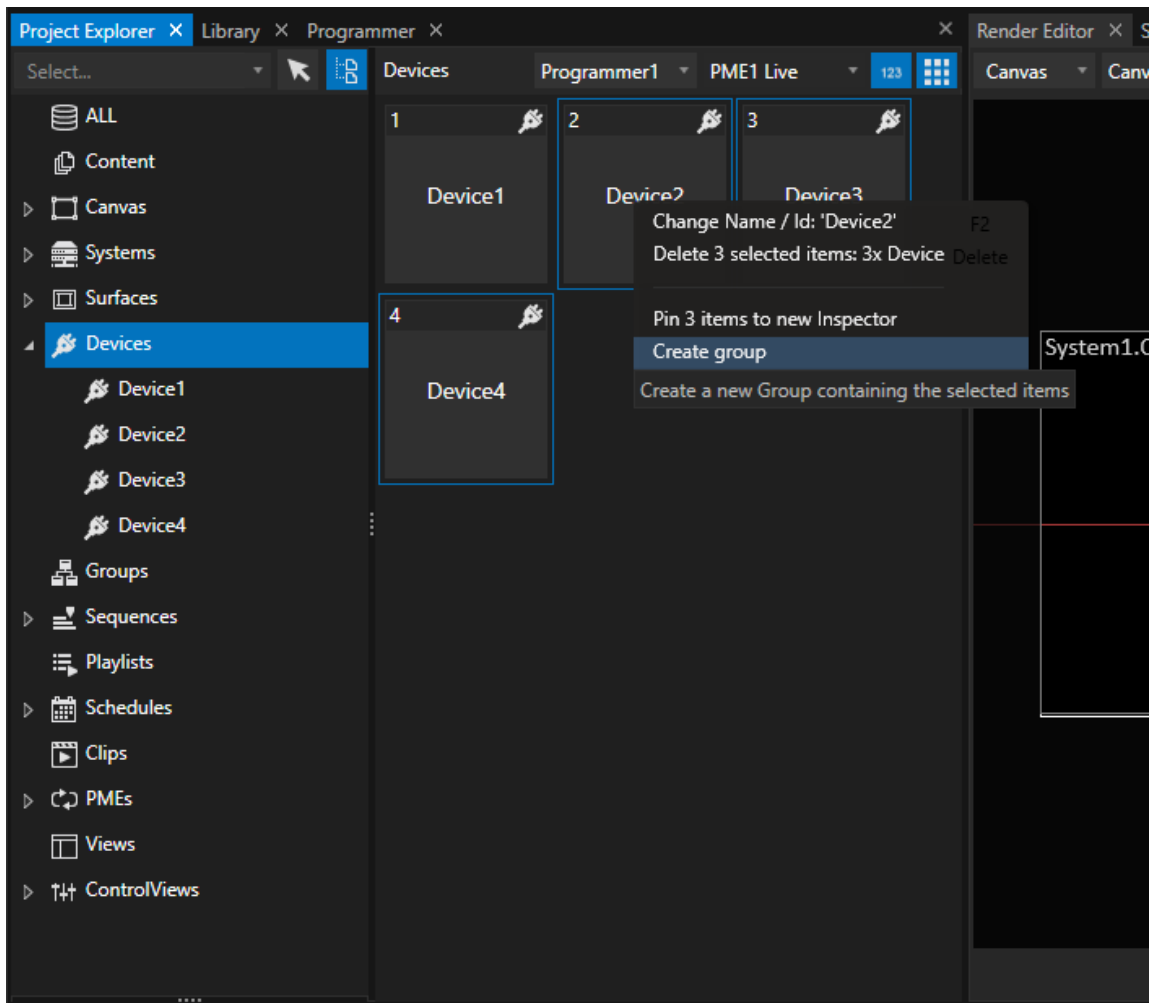
## Groups

- Manage and summarize your DMX Output Devices in Groups. Make value changes for all devices of a group

### Create a Group

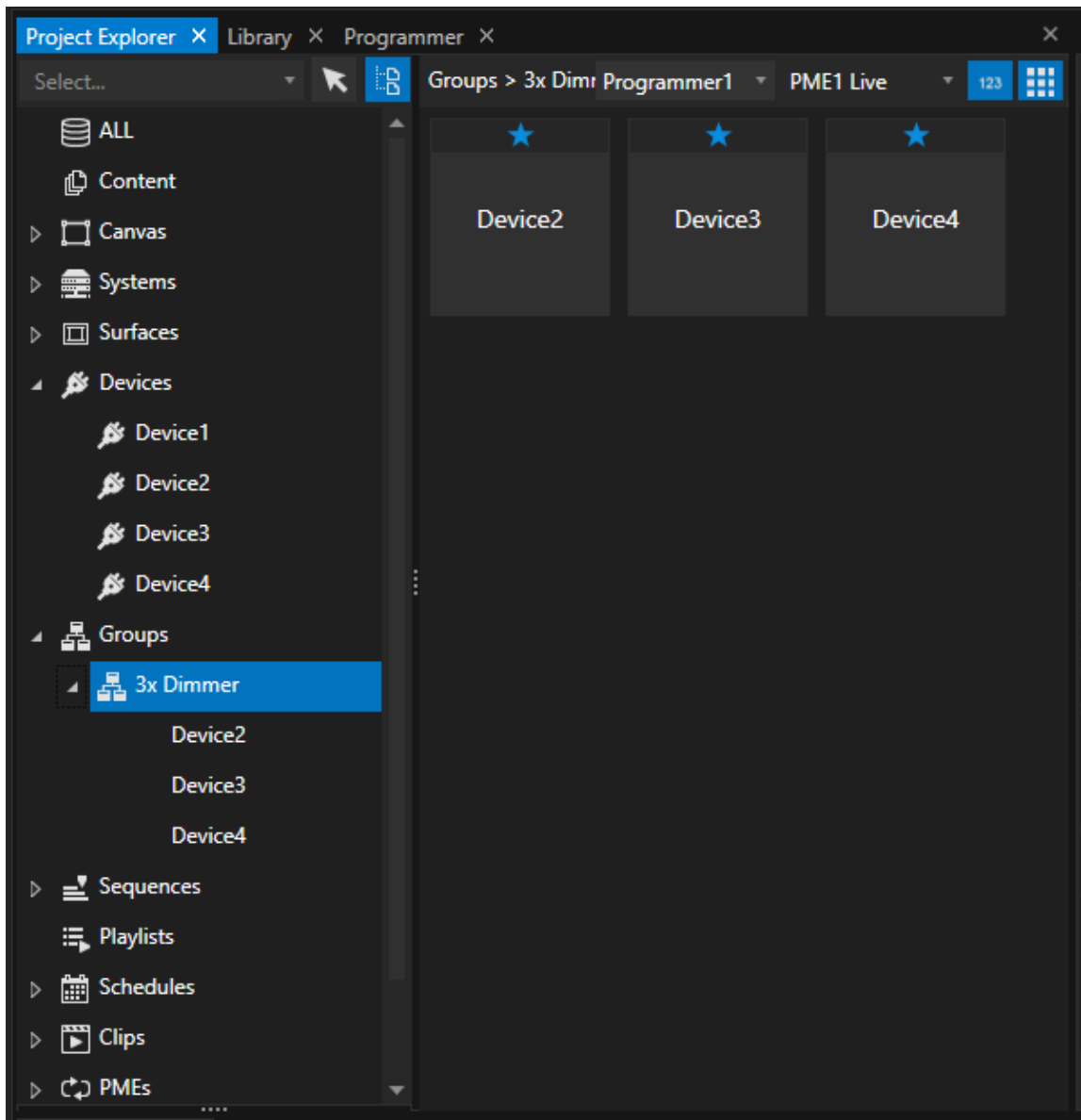
**There are 2 ways to create a group and assign Devices to it:**

- Go to the Device section into the Project Explorer
- Mark all devices that should be member of a group
- Right-Click and select "Create Group"
- A new Group that contains the selected Devices is created. You can find, select or rename the group in the "Group" section of the Project Explorer



or:

- Go to the "Groups" section in the Project Explorer
- Right-Click and open the context menu
- Choose "Add new" there
- Drag devices from Device section of the Project Explorer to this group

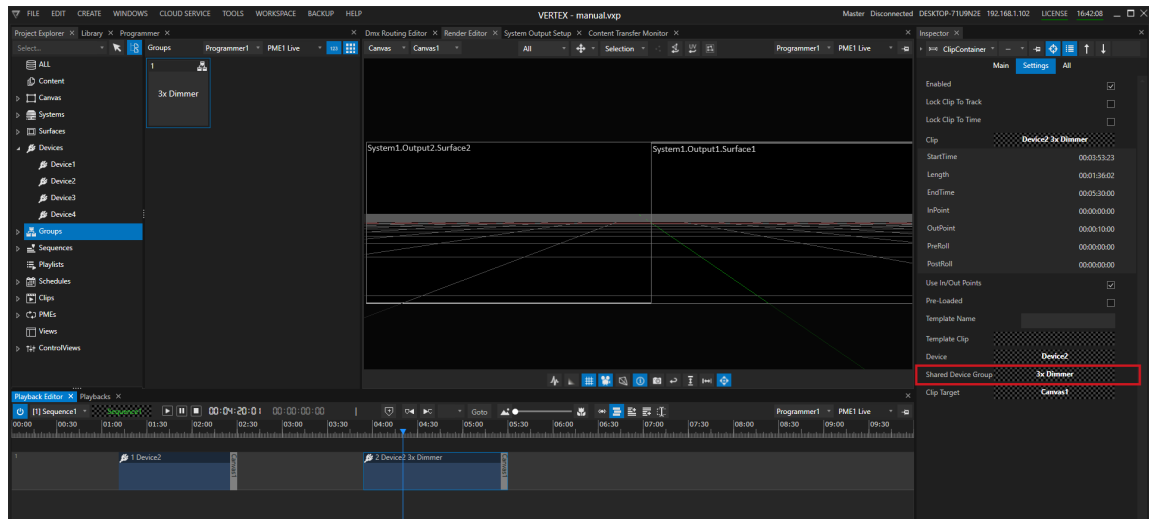


## Work with a Group of DMX-Devices

*When working with a group, you are able to set DMX Values for all group members with only one Clip Container.*

- Drag a Group from Project Explorer to the Playback Editor
- All changes you do for this Clip Container are adopted for all devices of this Group
- If values apply to a group, you can identify this with the "SharedDevice Group" Property. If there is a group set, all values of the clip container are also assigned to the group members.

- You can also transfer the settings for a DMX Device to a Group afterward by dragging a Group from Project Explorer to the "SharedDevice Group" field of an already existing Clip Container (that already hosts a DMX Device)



### 6.6.3 Powerpoint Remote

- VERTEX is able to **control a Microsoft Powerpoint application remotely** over network
- A **background service** is able to start Microsoft Powerpoint, load a .pptx-file, jump to a slide or set a presentation into full screen
- The background service can be controlled by a **PPT Remote Device** that is part of the **VERTEX device library**

## Use Cases

- You need to integrate Powerpoint with its full functionality into your project and not even import and read Powerpoint files
- Use VERTEX as master control system to remote start and control a PC with Powerpoint in your installation
- Capture the output of a Powerpoint PC as Live Input into VERTEX and control slide changes from your VERTEX timeline
- Read out notes of a Powerpoint file and display this notes as text on another screen - e.g. backstage

## What you need

- a PC with Microsoft Powerpoint installed and Microsoft Windows as operating system
- a PC with VERTEX installed
- a network connection between both PCs (Wireless LAN or cabled LAN)



### **VERTEX and Powerpoint on the same PC**

For Tests or even smaller projects with only one PC, you are even able to remote control a Microsoft Powerpoint application on the same PC on which VERTEX is running. Just use your local IP address as IP address of the PPT Remote Device

## Setup

### 1. Prepare Powerpoint PC

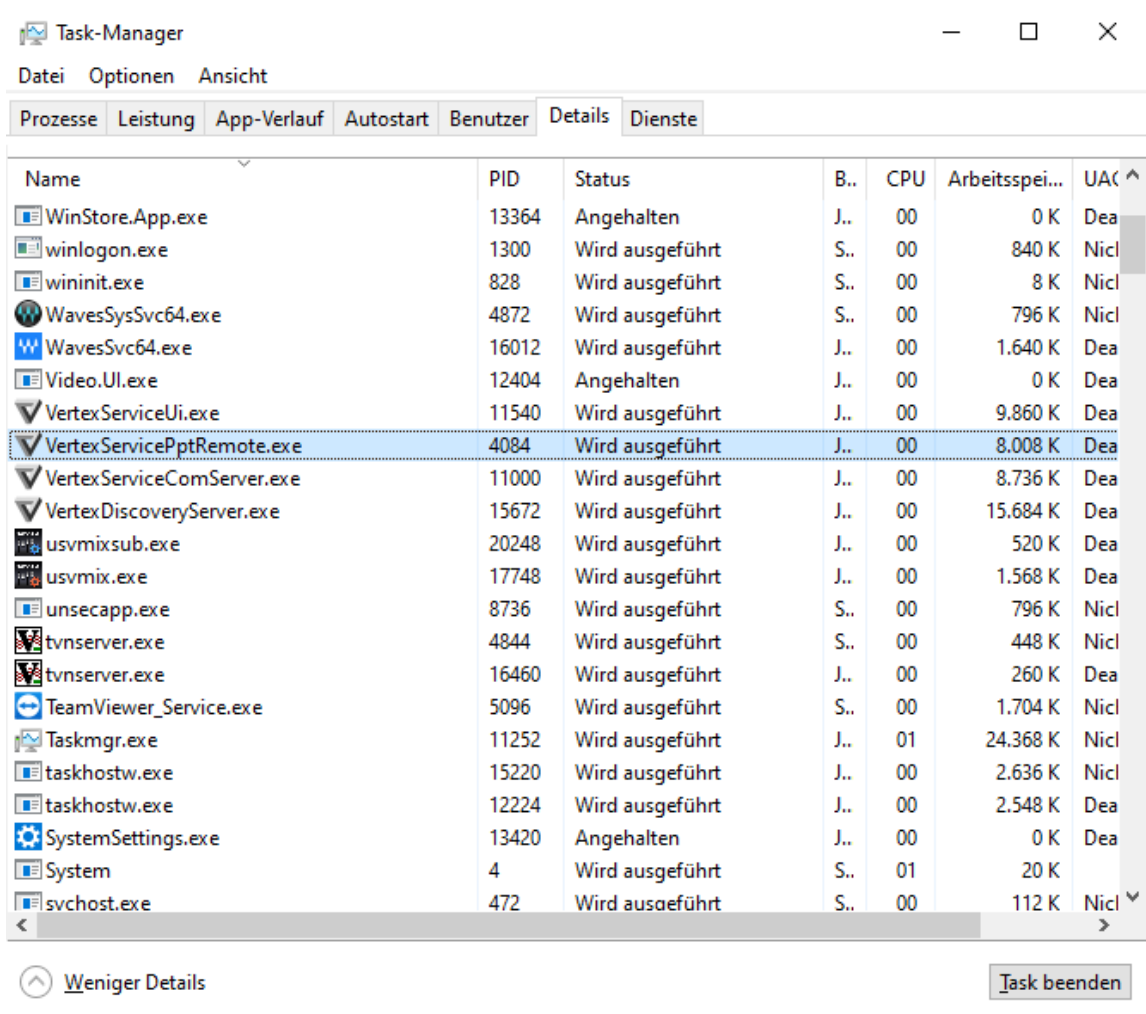
- **Install Vertex on the PC where Microsoft Powerpoint already is installed.**
- **Check your Network settings**, prefer a static IP address to avoid address changes after a reboot



### **VERTEX Trial Version is enough**

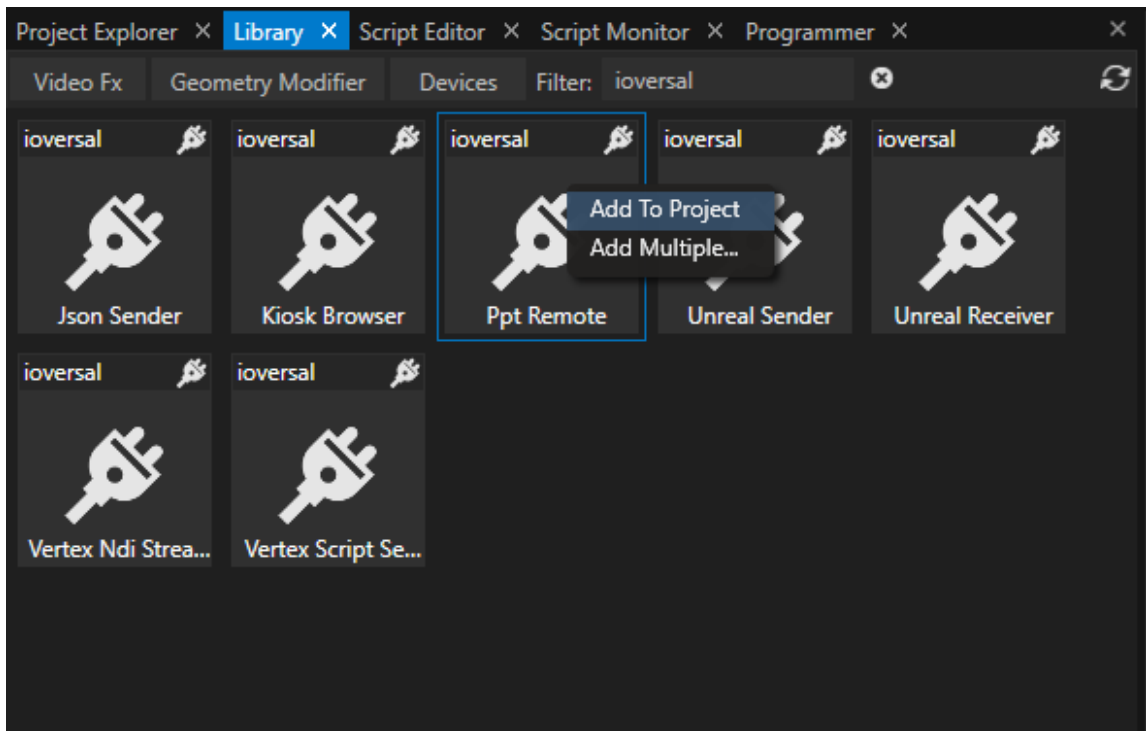
You don't need any Vertex license on the Powerpoint PC. The Vertex Trial version is enough. The Vertex installer will install a Windows background service to remote control Microsoft Powerpoint on this PC.

- **Reboot** the PC after installation
- Open Windows Task Manager and Check if `VertexServicePptRemote.exe` is running into background
- You don't have to start the main Vertex application on this computer. To remote control Powerpoint it is only necessary that all VERTEX background services are running.

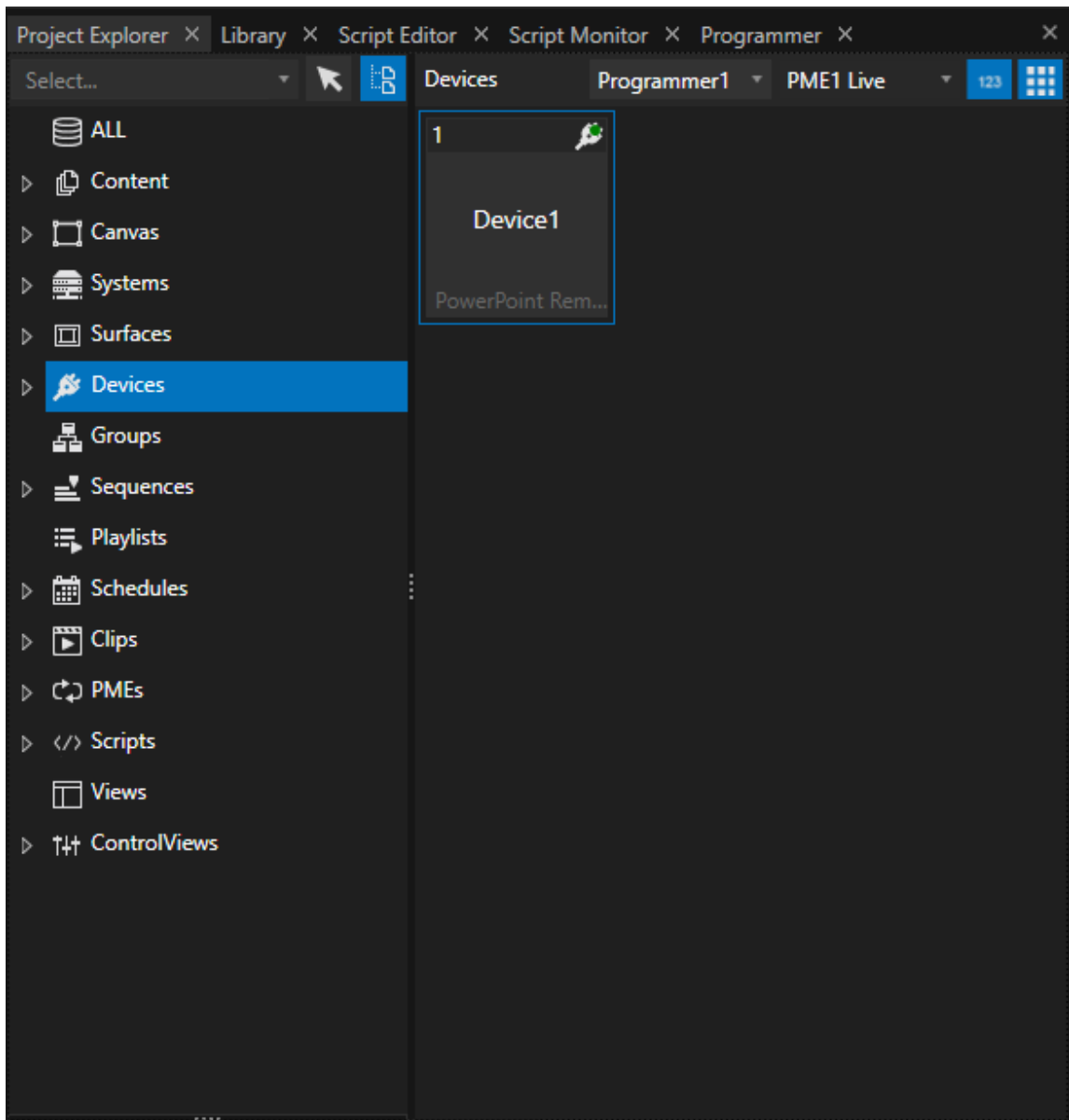


## 2. Prepare your VERTEX Project

- Search for the **"Ppt Remote"** device into the [Library Editor](#)
- **Open the context** menu by a right click
- **Add** this device to your project

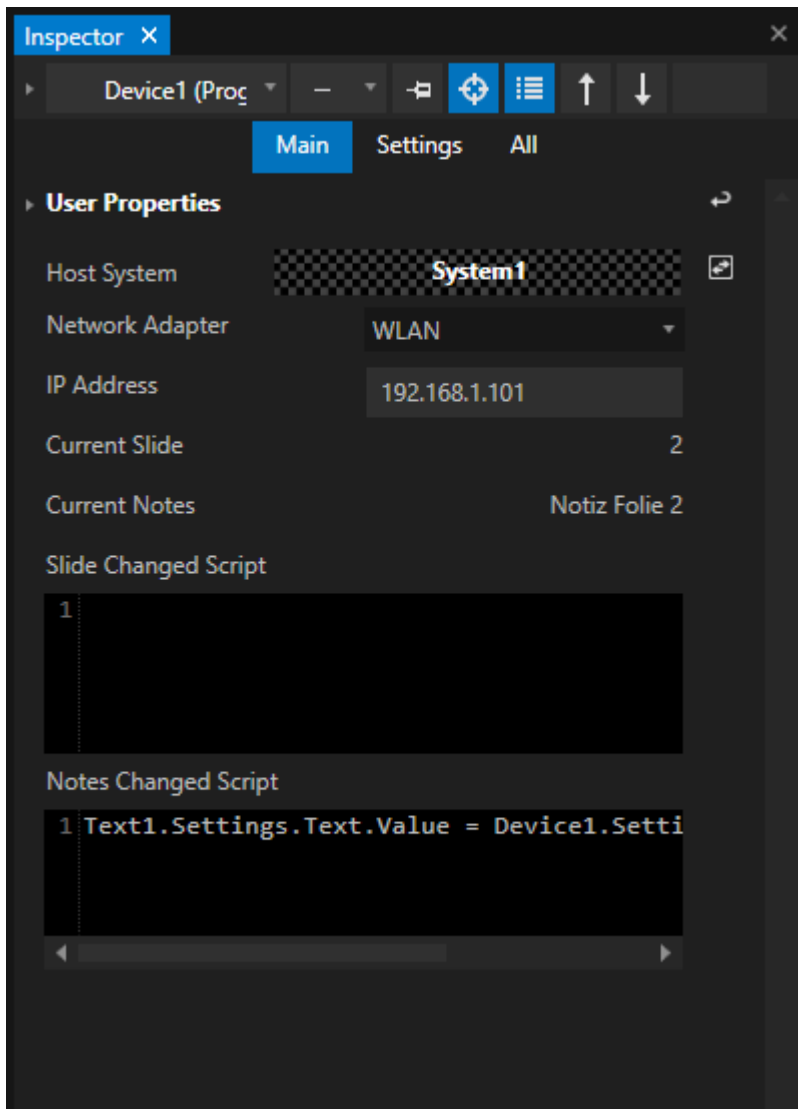


- Select the device into Project Explorer



- First configure the **Network Adapter of the Ppt Remote Device** in the **Inspector**
- Use the reconnect button and test the connection





- Now the Device should be ready to work with

## Examples

*Launch a specific Powerpoint file and go into presentation mode*

```
Device1.Launch C:\Users\User1\Desktop\test.pptx
```

*Quit whole Microsoft Powerpoint application*

```
Device1.Quit
```

**End presentation mode of Powerpoint but keep Powerpoint application running**

```
Device1.EndPresentation
```

**Go to Slide 2**

```
Device1.GotoSlide 2
```

**Example for a more complex start script**

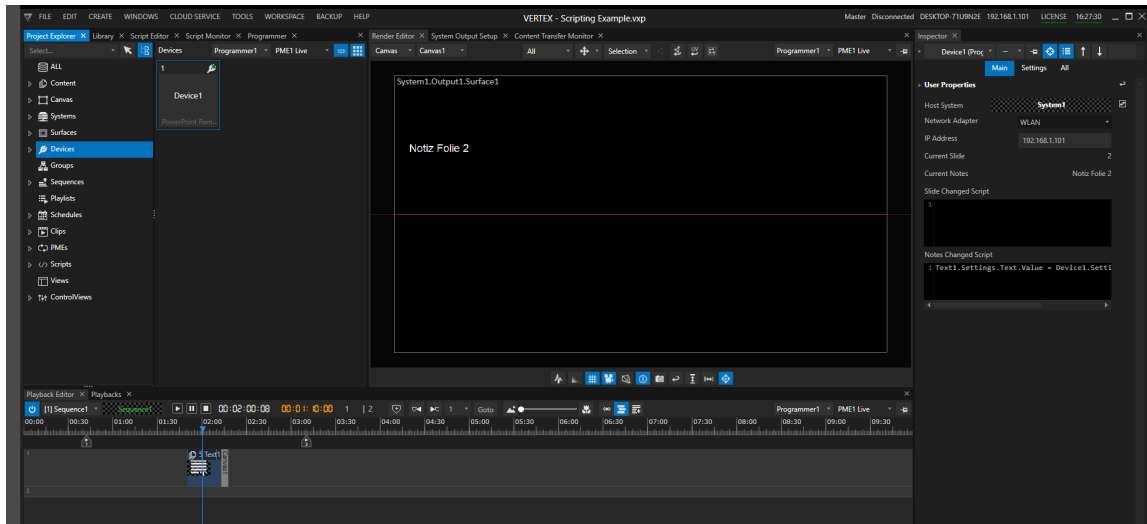
- Reset Connection (to be sure that connection is ready and working) and wait 1 second
- Set Volume of Powerpoint PC to 0 and wait 1 second
- Launch test.pptx file from Desktop of User1 and wait 3 seconds
- Go to next slide

```
Device1.ResetConnection  
wait 1  
Device1.SetVolume 0  
wait 1  
Device1.Launch C:\Users\User1\Desktop\test.pptx  
Wait 3  
Device1.NextSlide
```

**Show Notes from Powerpoint file as text content into VERTEX**

- The current powerpoint notes should be displayed as text content into VERTEX
- For our usecase, the text content into VERTEX should only be changed when the notes of the pptx are changed
- The Ppt Remote Device offers 2 options: "Slide Change Script" and "Notes Changed Script". For our usecase, add the following code snippet as Notes Changed Script

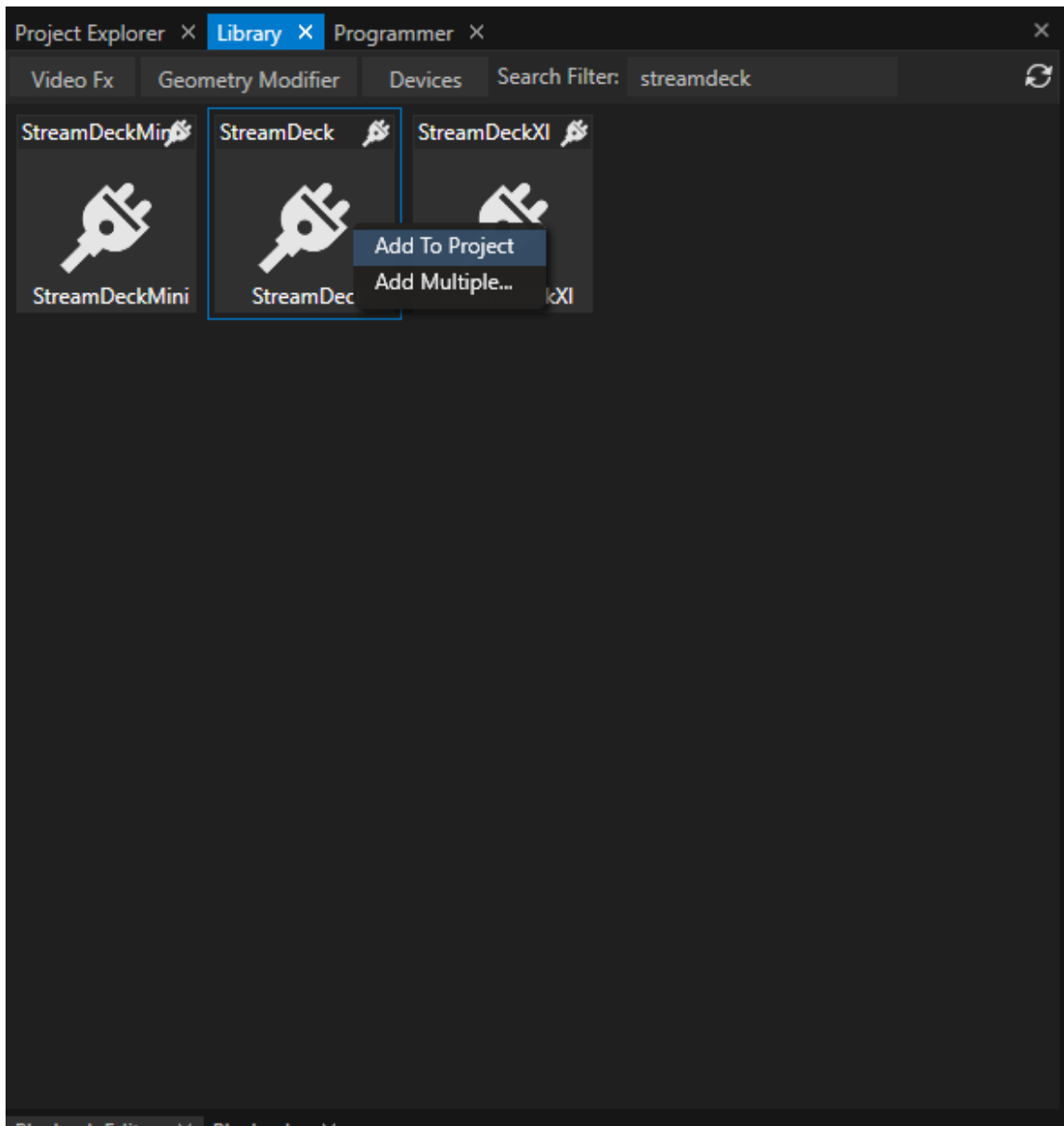
```
Content1.Settings.Text.Value = Device1.Settings.CurrentNotes.Value
```



## 6.6.4 Stream Deck

- VERTEX is shipped with **device Library elements** for the **Elagato Stream Deck, Stream Deck XL and Stream Deck mini**
- **No Driver installation** is required, just plug in the Stream Deck and start VERTEX
- **Settings for all Buttons** are done in VERTEX - **Script Commands** are used to trigger actions

## How to add a Stream Deck



1. **Connect the Stream Deck** to your PC#

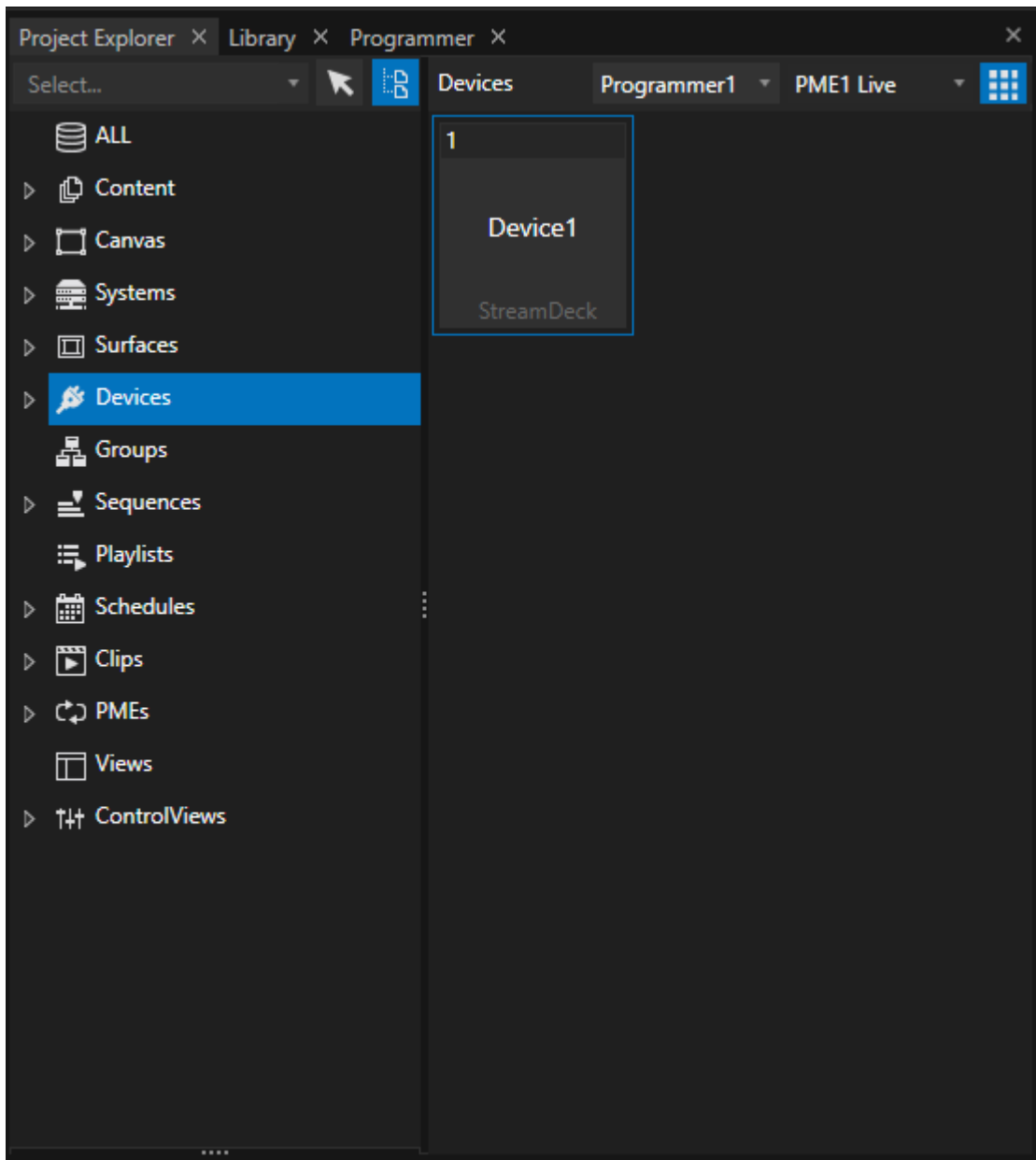


**Close Elgato "Stream Deck for Windows" Application and all other Applications that access your Stream Deck**

VERTEX directly accesses and talks to your Stream Deck device. Avoid side effects from other applications - like overwritten button designs - by closing all other applications before.!

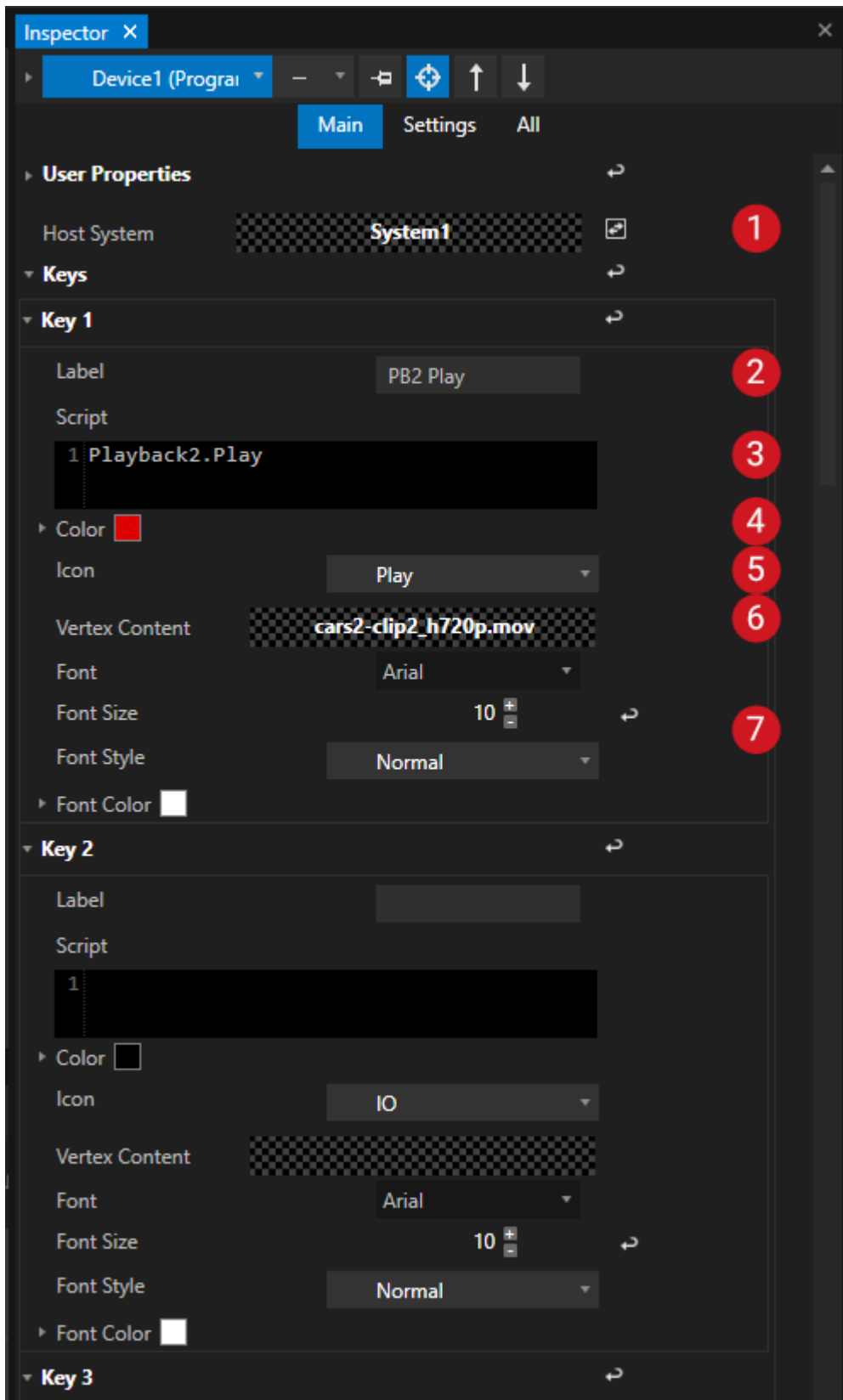
Especially the "Stream Deck for Windows" application from Elgato influences a Stream Deck that is used within VERTEX. Please also terminate this application before you start with your VERTEX project.

2. **Start VERTEX**
3. Open **Library Editor** and search for the **Stream Deck Device**
4. Right-click with you mouse on the Device and **add it to your Project**
5. **Select the device** in the **Project Explorer**
6. Go to the **Inspector and set the values** for each button



Your Stream Deck device and all settings are stored into your VERTEX project file. When you load the Project, just connect the Stream Deck to your PC and go ahead with your work.

## Settings



<b>1</b>	<b>Host System</b>	Defines the host System into your Project <b>where your Stream Deck physically is connected to.</b> Only change when working into a project with Session Member Systems.
<b>2</b>	<b>Label</b>	Set a <b>Label</b> that is <b>displayed as text</b> for your Stream Deck button.
<b>3</b>	<b>Script</b>	Enter <b>Script Commands that should be executed when the Stream Deck Button is pressed</b> Focus with your Mouse and use Shortcut "CTRL + Space"to show a list of all Commands
<b>4</b>	<b>Color</b>	Sets a <b>background color</b> for the Stream Deck <b>button.</b>
<b>5</b>	<b>Icon</b>	<b>Choose an icon for your button. VERTEX comes up with predefined Icons for the most common use cases</b>
<b>6</b>	<b>Vertex Content</b>	Show a <b>thumbnail from a content asset of your project as background of the button.</b> Drag an video or image content with your mouse from Project Explorer to this property field Set Icon to "None". Right-click and open context menu to clear background image
<b>7</b>	<b>Font Settings</b>	<b>Font settings for the button Label</b>



#### **Button Count**

The Stream Deck buttons are counted in rows, starting from top left

## Further Settings

- Select the "Settings"-Tab of the Inspector to set the global brightness of the Stream Deck or to check if the Stream Deck is connected to your PC.

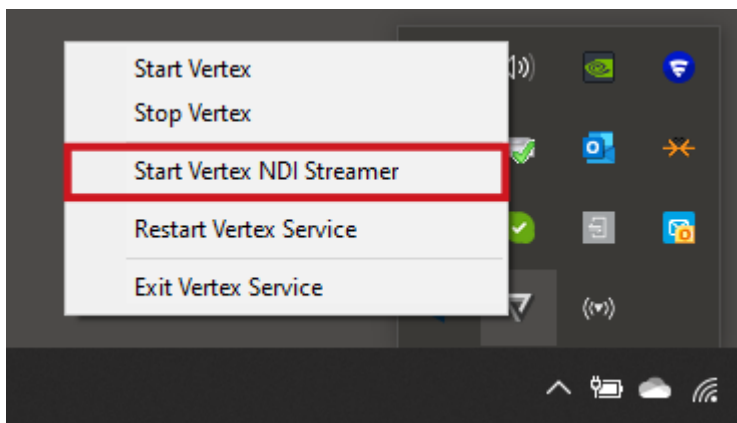


## 6.6.5 Vertex NDI Streamer

- VERTEX NDI Streamer is a **multi source NDI toolkit**
- VERTEX NDI Streamer is **capable to stream up to 2 desktop regions and 1 live input** as a NDI Stream
- The NDI Streamer **is free** and fully **works without a valid VERTEX license**
- **In addition, all settings and streams can be controlled remotely out of VERTEX (with a license)**

### Start NDI Streamer

- The VERTEX NDI Streamer comes up in a package with the VERTEX installer
- Just download and install an up-to-date Version of VERTEX
- Right-click with your Mouse on the VERTEX Tray Icon on the Windows taskbar
- Start Vertex NDI Streamer

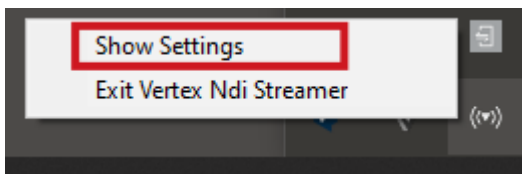


- When the NDI Streamer is started, a new Icon appears into the windows tray bar



## Set up NDI Streams

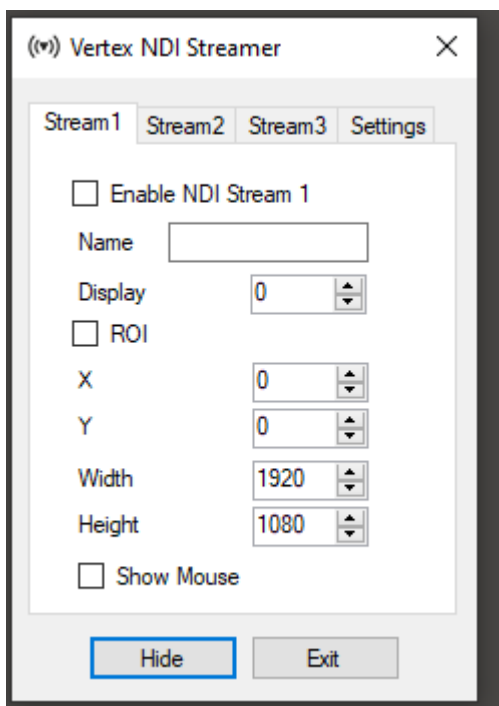
- Right-Click on the VERTEX NDI Streamer tray icon in the windows taskbar
- Select "Show Settings"



- Now you can enable up to 3 NDI Streams from your PC

### Stream 1 + 2

enable you to set up two NDI streams that capture a display (or only a region) of your PC.



#### **Name**

Name of the stream that is shown for all NDI applications

**Display**

Select the display number of your PC that should be sent as NDI stream.

**ROI**

Region of Interest. Default: disabled

Enable to capture only a region of the selected display as NDI stream

Enter pixel coordinates (starting at top left) and width/height to define the ROI

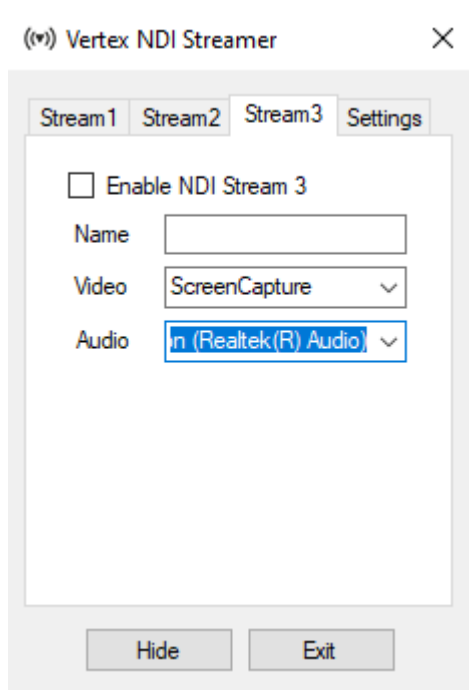
**Show Mouse**

Show mouse pointer in NDI stream

## Stream 3

Stream 3 differs from the first ones. Here, you are able to send a video source (e.g. a webcam) and audio source (like a microphone)

from your Windows PC as NDI stream.

**Name**

Name of the stream that is shown for all NDI applications

**Video**

Select from a list of video sources.

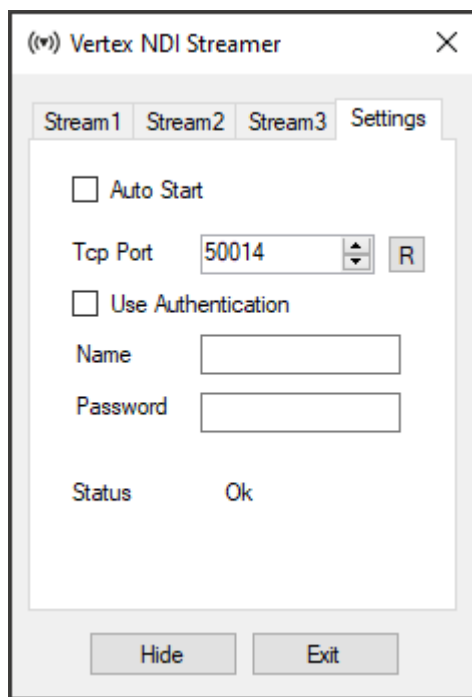
Depending on your PC hardware and configuration the list of available sources may vary

### Audio

Select from a list of audio sources.

Depending on your PC hardware and configuration the list of available sources may vary

## Settings



### Auto Start

Enable to add the NDI Streamer to Microsoft Windows autostart

### TCP Port

Change port number for remote controlling the NDI streamer over TCP.

Use the R-Button to restart the TCP Servers after a port change.

### R Button

Reset and Restart the TCP Server for all network adapters after you have changed the TCP Port

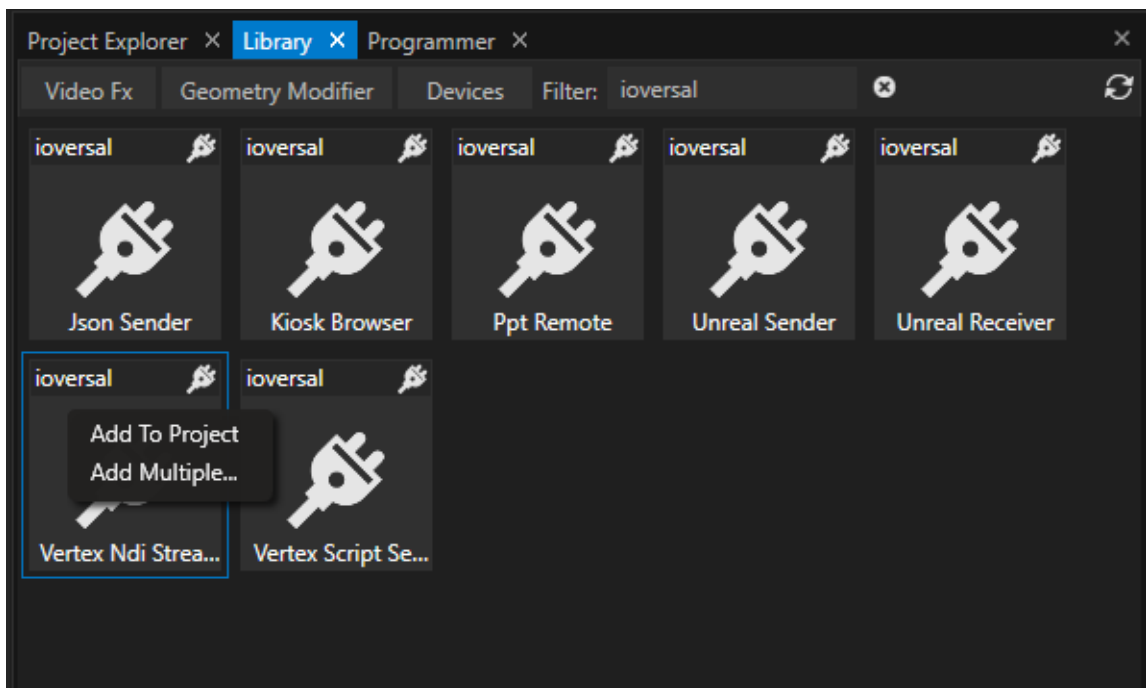
### Use Authentication

Set User Name and Password for TCP Authentication

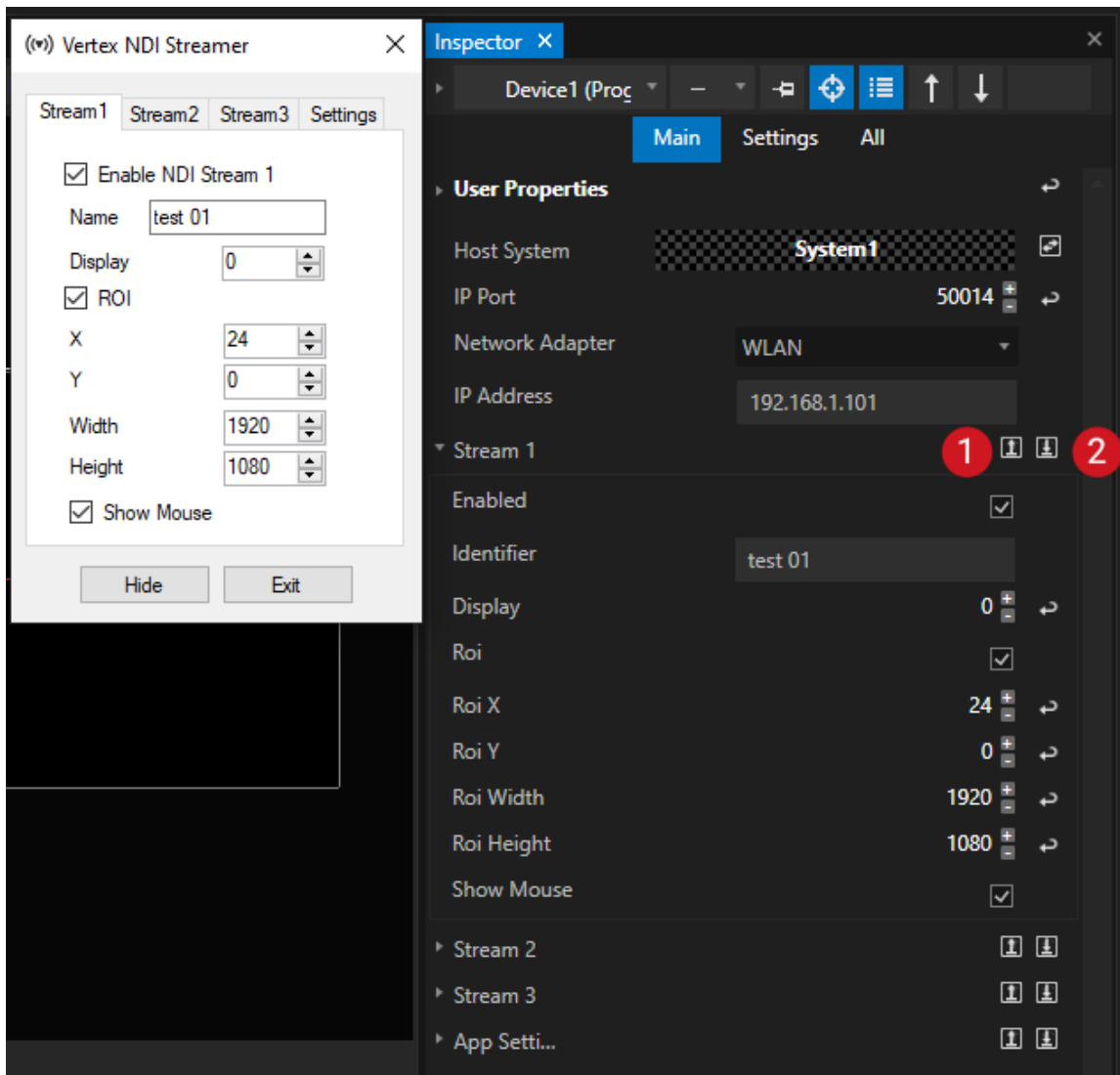
## Remote Control out of VERTEX

- You can **remote control all settings of the VERTEX NDI Streamer** on a PC in your network **out of VERTEX**
- Just add a **"Vertex NDI Streamer" Device** from the [Library](#) into your current project
- Once configured, you have **access to all settings of the NDI Streamer application** and are able to automate your NDI workflow

### Setup



- Open the [Library](#)
- Select **Devices** and filter for **"ioversal"**
- Right-click to the device **"Vertex NDI Streamer"**
- **"Add to project"**
- Select the device in the [Project Explorer](#) and set the initial properties in the [Inspector](#)
- Once the connection is set up, you are able to upload and download all application and stream settings.



Comparison: All settings from the NDI Streamer application (on the left) you will also find into the Inspector when selecting the corresponding Device "Vertex NDI Streamer" (on the right)

**IP Port**

Enter here the same IP Port as set in the NDI Streamer application

**Network Adapter**

Choose a network adapter from your local VERTEX instance

**IP Address**

Enter IP address of the PC on that the Vertex NDI Streamer application is running and should be remote controlled

**Upload Settings** 1

Upload all current settings from VERTEX to the NDI Streamer application

### Download Settings 2

Download all current settings from NDI Streamer application to Inspector

## Scripting

As with every Device in VERTEX, you can control an NDI Streamer Device with a [Script](#).  
For the following examples the NDI-Streamer Device was added as "Device1" to a project

### Enable and disable Stream 1 on the remote PC

```
Device1.Settings.Stream1.Enabled.Value = true  
Device1.Settings.Stream1.Enabled.Value = false
```

### Show Mouse Cursor in NDI Stream 1

```
Device1.Settings.Stream1.ShowMouse.Value = true
```

## 6.7 DMX-512

- VERTEX offers you **different input and output options** to work with **DMX-512** into your project
- All **popular Ethernet-based protocols for DMX-data are supported**: Art-Net™, sACN or DMX-512 with ioversals <%DMX\_IO>- Interface. Additionally MA-Net2 is supported only as Input
- With the **DMX-Routing Editor** you define the incoming or outgoing DMX-protocol and set the universe routing.

### Create a DMX-Routing and define the Input and output settings

Your work into the DMX-World always should start with a DMX-Routing, where you define the Input and output settings and protocols:

[DMX Routing](#)

ioversal offers you an driverless USB-DMX Interface - the DMX io to work with DMX-512 into VERTEX. Read more about the configuration:

[Dmx IO](#)

### Output DMX Data

[Use Devices to Output DMX-Data.](#)

[Route incoming protocols into a VERTEX network and output it on another System](#)

### Use Incoming DMX-Data to Control

There a **different options** to control VERTEX **with DMX-512**.

Choose between or mix the following ones:

[Control a Playback with DMX-512](#)

[Control a System with DMX-512](#)

[Manipulate and Control a Surface with DMX-512](#)

[Use VERTEX like a digital lighting Fixture with Console Layers](#)

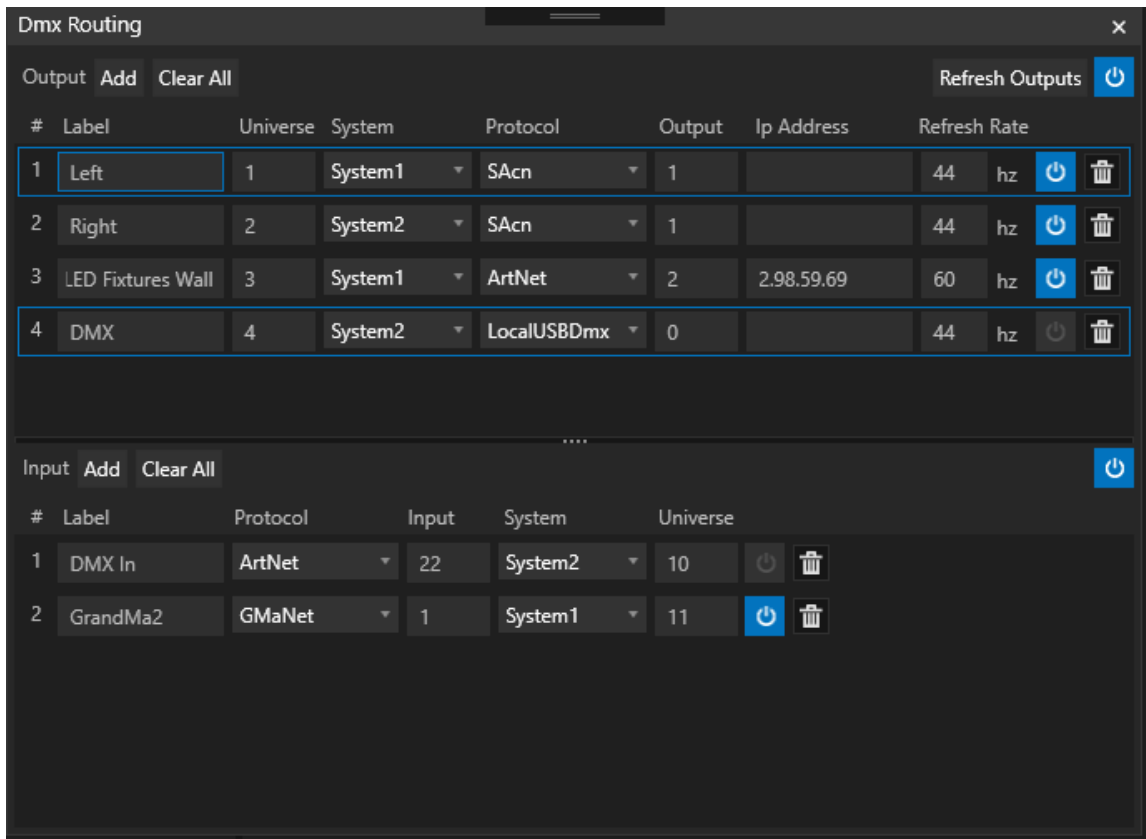
[Use DMX-Input Devices for scripts, conditions and/or triggers based on incoming DMX-Data](#)

### 6.7.1 DMX Routing

- The DMX-Routing Window is the place for a **central management** of **all DMX-over-Ethernet Inputs and Outputs** for your project



- Vertex works with an **internal decimal universe count** - that gives you the freedom to map, route and split those universes to different protocols, output universe numbers and Systems. Until your final patching, the internal programming in VERTEX **is independent from your final output** routing.
- Manage **all your Systems** in one window and **split** the physical **Ethernet output** up to **other Systems** of your project.



## Output

1	Label	Free Label Name
2	Universe	Decimal VERTEX Universe number: VERTEX is working with an internal and decimal Universe Count for Lighting Control Data. Independent from e.g. the DMX Start address of your device, you could give all devices a Universe number that stays consistent in your whole project
3	System	Defines the System on which Ethernet Card the control data should be sent. For e.g different rooms or room situations, you could e.g. send out 2 Universes from your Master System. For the Art-Net™ Control data for

		<i>e.g. some LED-Fixtures you could use the network card and network from your client System2 that is e.g located in another area...</i>
4	<i>Protocoll</i>	<i>Select the Protocol for your Lighting Control data. You can select for every universe between Art-Net™, sACN or DMX over a connected USB-DMX-Output device</i>
5	<i>Output</i>	<i>Defines the universe, on which your control data will be send out. So you could e.g route and output VERTEX Universe 1 to Art-Net™ Universe 1, or even VERTEX Universe 1 to sACN Universe 2</i>
6	<i>IP-Address</i>	<i>By Default, if there is no entry, Art-Net™ data will be broadcasted and sACN-Data will be multicasted. When you enter an IP address into this field, Art-Net™ or sACN -Data will be unicasted to the device with this IP.</i>
7	<i>Frames per Second</i>	<i>Per default this field is set to 44,1 Hz (DMX-Standard). You can individually increase or decrease the protocol frames per second by entering your custom values. The maximum number is limited to 100, in most cases the ethernet-setup limits the maximum to around 80 frames.</i>
8	<i>Single Mute</i>	<i>Disables the network connection of a single column</i>
9	<i>Global Mute</i>	<i>Disables all DMX-over-Ethernet connections</i>

## Input

7	<i>Label</i>	<i>Free Label Name</i>
8	<i>Protocol</i>	<i>Input protocoll on that VERTEX will listen</i>
9	<i>Input</i>	<i>Input Universe of the lighting protocol that VERTEX should listen/Read</i>
10	<i>System</i>	<i>VERTEX System on your Project Network that should read/listen to the incoming protocoll. This System should have physical access to the protocoll, of course ..</i>
11	<i>Universe</i>	<i>VERTEX Universe on which the incoming control data is mapped</i>

## Supported Protocols

### Output Protocols:

- Art-Net™
- sACN
- DMX-512 over USB-Adaptor

### Input Protocols:

- Art-Net™
- sACN
- MA-Net2, could only be received as Input and not send out as output protocol
- DMX-512 over USB-Adaptor



#### Good to know: Broadcast, Unicast, Multicast

Art-Net™ supports per default specification broadcast or unicast.

sACN supports per default specification unicast or multicast. Please note: multicast is very effective for a larger amount of control data into a network.

When planning to sending data with multicast: Please be aware that your network-switch has to support this. Not all hardware does.

### Advanced: Universe Numbering with Art-Net™ and sACN

- Within the **VERTEX [DMX Patching](#)**, we are using a **decimal numbering** for all control universes. This includes our internal numbering as well as the Art-Net™ and sACN numbering.
- **By default, Art-Net™ is defined by using a combination of subnets and universes within hexadecimal numbering.** To make it easier to switch between output protocols and to simplify your patching, **VERTEX internally uses only decimal numbers** for ethernet based Control-Data.
- The **output for Art-Net™ matches of course to the protocol specifications.** VERTEX takes over the conversion into subnet and universes for you under the hood.

## Comparison between hexadecimal and decimal numbering

VERTEX Universe	sACN Universe Numbering	Default Art-Net™ Universe Numbering
-----------------	-------------------------	-------------------------------------

Numbering (Decimal) for interal as for ethernet based control universes		with subnets and universes (Hexadecimal)	
		subnet	universe
0	0 is reserved by default and not used	0	0
1	1	0	1
2	2	0	2
3	3	0	3
4	4	0	4
5	5	0	5
6	6	0	6
7	7	0	7
8	8	0	8
9	9	0	9
10	10	0	A
11	11	0	B
12	12	0	C
13	13	0	D
14	14	0	E
15	15	0	F
16	16	1	0
17	17	1	1
18	18	1	2
19	19	1	3
20	20	1	4
21	21	1	5
22	22	1	6

23	23	1	7
24	24	1	8
25	25	1	9
26	26	1	A
27	27	1	B
28	28	1	C
29	29	1	D
30	30	1	E
31	31	1	F
32	32	2	0
33	33	2	1
34	34	2	2
35	35	2	3
36	36	2	4
37	37	2	5
38	38	2	6
39	39	2	7
40	40	2	8
41	41	2	9
42	42	2	A
43	43	2	B
44	44	2	C
45	45	2	D
46	46	2	E
47	47	2	F
48	48	3	0
49	49	3	1
50	50	3	2
51	51	3	3

52	52	3	4
53	53	3	5
54	54	3	6
55	55	3	7
56	56	3	8
57	57	3	9
58	58	3	A
59	59	3	B
60	60	3	C
61	61	3	D
62	62	3	E
63	63	3	F
64	64	4	0
65	65	4	1
66	66	4	2
67	67	4	3
68	68	4	4
69	69	4	5
70	70	4	6
71	71	4	7
72	72	4	8
73	73	4	9
74	74	4	A
75	75	4	B
76	76	4	C
77	77	4	D
78	78	4	E
79	79	4	F
80	80	5	0

81	81	5	1
82	82	5	2
83	83	5	3
84	84	5	4
85	85	5	5
86	86	5	6
87	87	5	7
88	88	5	8
89	89	5	9
90	90	5	A
91	91	5	B
92	92	5	C
93	93	5	D
94	94	5	E
95	95	5	F

**Universes 96-191**

<i>VERTEX Universe Numbering (Decimal) for interal as for ethernet based control universes</i>	<i>sACN Universe Numbering</i>	<i>Default Art-Net™ Universe Numbering with subnets and universes (Hexadecimal)</i>	
		<b>subnet</b>	<b>universe</b>
96	96	6	0
97	97	6	1
98	98	6	2
99	99	6	3
100	100	6	4
101	101	6	5

102	102	6	6
103	103	6	7
104	104	6	8
105	105	6	9
106	106	6	A
107	107	6	B
108	108	6	C
109	109	6	D
110	110	6	E
111	111	6	F
112	112	7	0
113	113	7	1
114	114	7	2
115	115	7	3
116	116	7	4
117	117	7	5
118	118	7	6
119	119	7	7
120	120	7	8
121	121	7	9
122	122	7	A
123	123	7	B
124	124	7	C
125	125	7	D
126	126	7	E
127	127	7	F
128	128	8	0
129	129	8	1
130	130	8	2



131	131	8	3
132	132	8	4
133	133	8	5
134	134	8	6
135	135	8	7
136	136	8	8
137	137	8	9
138	138	8	A
139	139	8	B
140	140	8	C
141	141	8	D
142	142	8	E
143	143	8	F
144	144	9	0
145	145	9	1
146	146	9	2
147	147	9	3
148	148	9	4
149	149	9	5
150	150	9	6
151	151	9	7
152	152	9	8
153	153	9	9
154	154	9	A
155	155	9	B
156	156	9	C
157	157	9	D
158	158	9	E
159	159	9	F

160	160	10	0
161	161	10	1
162	162	10	2
163	163	10	3
164	164	10	4
165	165	10	5
166	166	10	6
167	167	10	7
168	168	10	8
169	169	10	9
170	170	10	A
171	171	10	B
172	172	10	C
173	173	10	D
174	174	10	E
175	175	10	F
176	176	11	0
177	177	11	1
178	178	11	2
179	179	11	3
180	180	11	4
181	181	11	5
182	182	11	6
183	183	11	7
184	184	11	8
185	185	11	9
186	186	11	A
187	187	11	B
188	188	11	C

189	189	11	D
190	190	11	E
191	191	11	F

Art-Net™ Designed by and Copyright Artistic Licence Holdings Ltd

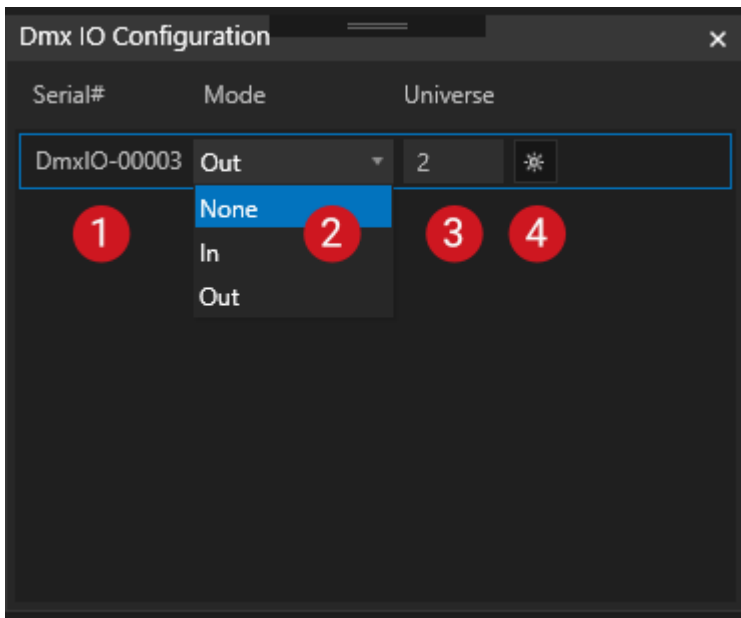
## 6.7.2 DMX IO

- As an **USB-DMX Interface**, the ioversal DMX IO can operate **either as DMX-Input** or as **DMX-Output**
- **No special Windows hardware driver is required** - just plug in the ioversal DMX IO into the USB-interface of your PC
- **Multiple interfaces could operate independently on one PC** or in a Vertex network



## Set-up and Configuration

1. Plug a ioversal DMX IO into a USB-interface of your PC (both USB 2 or USB 3 are supported)
2. You don't need to install any specific driver, Windows 10 detects the USB device and uses a standard USB driver for generic devices
3. The Status-LED of the ioversal DMX IO should glow
4. Start Vertex and create a new project or open your recent project file
5. Go to the "Windows"- tab in the top main menu
6. Open "DMX IO Configuration" there



<b>1</b>	<b>Serial Number</b>	Unique Serial Number of the detected ioversal DMX IO Device
<b>2</b>	<b>Mode</b>	Mode Selection - Select between DMX In or OUT - this Mode is saved into the attached ioversal DMX IO. The Interface will remember this setting System independently when it is plugged in again -
<b>3</b>	<b>Universe</b>	Universe Settings For Mode = In: Select the VERTEX internal Universe, the incoming DMX Signal is routed to. For Mode = Out: Number of VERTEX-internal Universe that should send as DMX. The interface will hear for data on this universe number

		<i>This Number is saved into the attached ioversal DMX IO. The Interface will remember this setting System independently when it is plugged in again - Fore more details about routing, please also read: <a href="#">DMX-Patch</a></i>
<b>4</b>	<b>Identify</b>	<i>Highlight- Button for Identification: When pushing the Highlight-Button, the LED of the corresponding ioversal DMX IO- Interface will blink. Especially helpful when working with multiple Interfaces at the same PC.</i>



#### **Universe Number and Mode are internally stored in the interface**

You can Plug on and Off the ioversal DMX IO and they will work again in the same Mode and grabbing/sending out the same VERTEX Universe Number. Once configured, you are able to plug in the Interface on any System of your project and it will remember on your settings.

### 6.7.3 DMX Control

- There are **different options** to control VERTEX with **incoming DMX-Data**
- Just **control Playbacks** or **Systems** or make use of DMX-512 to make a more complex **Surface modulation**
- VERTEX additionally introduces the concept of a **Console Layer** that gives you the options to control **VERTEX like a lighting fixture**
- With a **DMX-Input Device** you are able to use DMX data into scripts or into e.g. the Trigger-Editor



#### **DMX Routing**

To make use of all options below, **you first have to create a [DMX Routing](#)**. Use the [DMX Routing Editor](#) to set up your incoming DMX protocol (Art-Net™, sACN, MA-Net2 or DMX-512 with ioversals ioversal DMX IO)

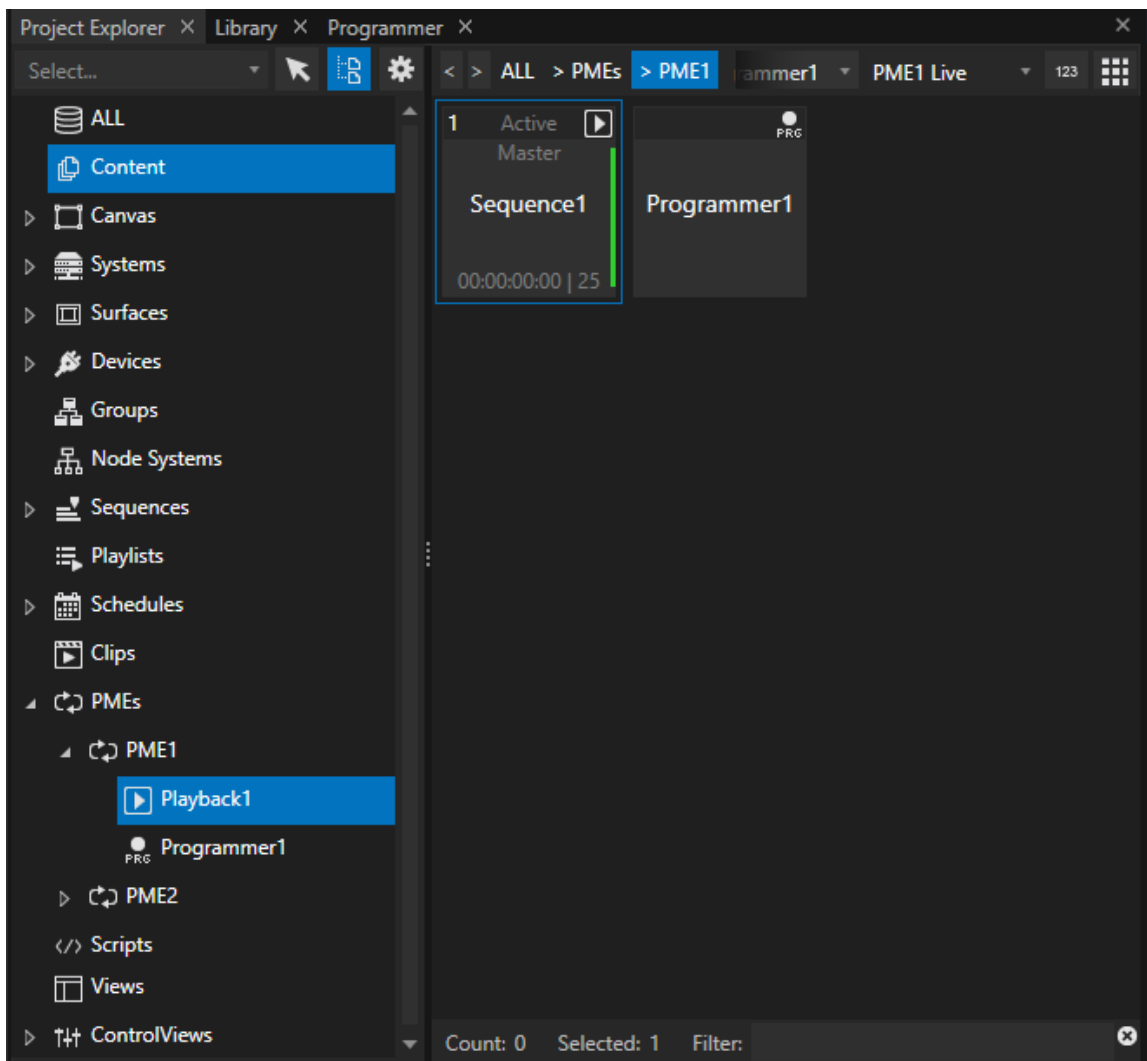
You have to do **this setup once**.

## Control a Playback with DMX-512

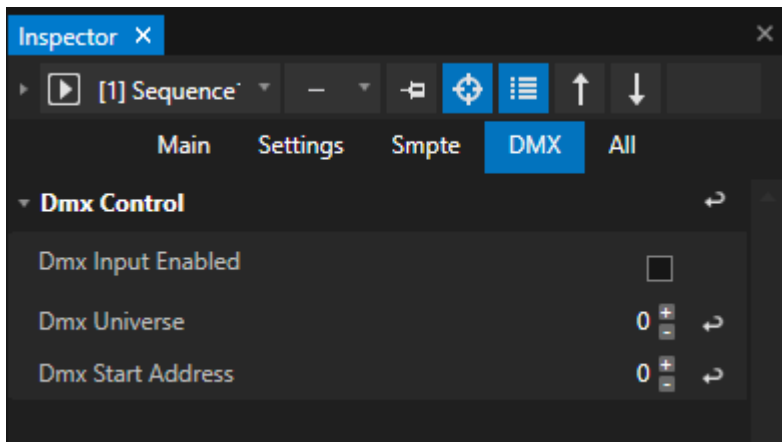
- Once a [DMX Routing](#) is created, VERTEX offers you the option to **control a Playback by incoming DMX-Data**
- With **8 DMX-Channels** your are able to control e.g. opacity, play/pause or jump to a cue
- DMX-512 **settings** for Playback control are **located into the PME** (Playback Mixing Engine).

## Settings

- Open PME section into Project Explorer
- Select PME1 (live)
- Select the Playback you want to control by DMX



- Select the DMX Tab into Inspector and do your settings for start address and universe
- Don't forget to activate the DMX Input for this Playback



### DMX Channel Map

DMX Address	Parameter Name	Type	Default Values	Values
1	Opacity	16bit	65535	0-65535
3	On/Off	8bit	255	0-127 Off    128-255 On
4	Play Pause Stop	8bit	45	0-10 Idle/None    11-20 PlayOnce    21-30 Play Loop    31-40 Pause    41-50 Stop
5	Speed Factor	8bit	128	0-124 0x - 1x    125-130 1x    131-255 1x - 4x
6	Cue Selection	16bit	0	0-65535
8	Frame Selection	16bit	0	0-65535

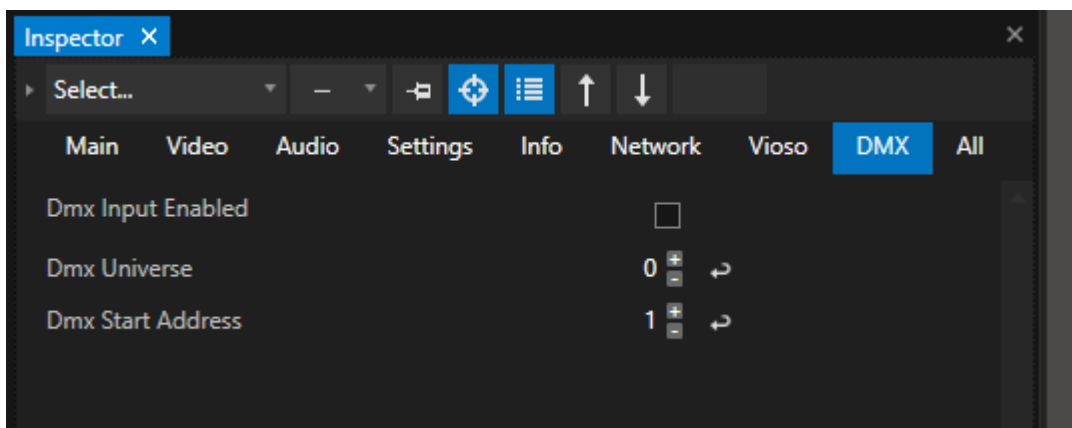
## Control a System with DMX-512

- Once a [DMX Routing](#) is created, VERTEX offers you the option to **control a System by incoming DMX-Data**

- With **6 DMX-Channels** your are able to control e.g. opacity, system volume, trigger Scripts by ID or set functionality

## Settings

- Select your System and go to Inspector



- Select the DMX Tab into Inspector and do your settings for start address and universe
- Don't forget to activate the DMX Input for this System

## DMX Channel Map

	Parameter Name	Type	Default Value	Note
1	Live Master	16bit	0	16bit opacity 1 mode for crossfade wipe modetexture
2	fine			
3	Live Volume	16bit	0	



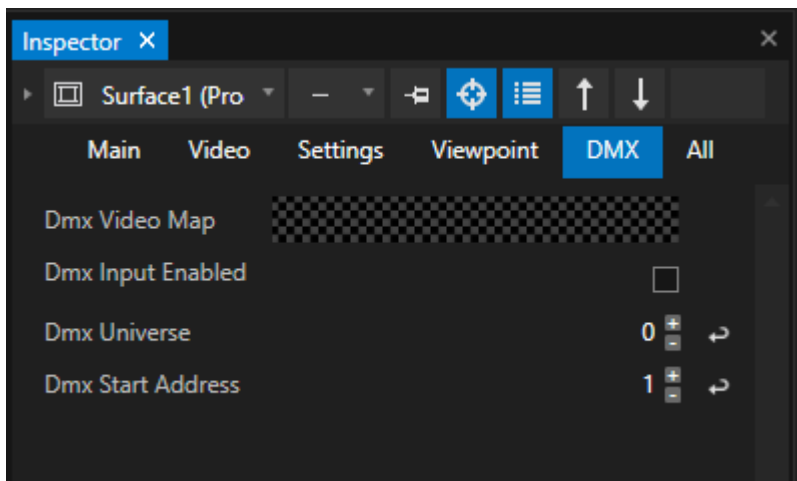
4	fine				
5	Script Control	8Bit	0		0=idle 1-254 Script Id 255 Take after 1sec
6	Function Control	8Bit	0		0=idle 255 Take after 1sec values: 1 Enter Fullscreen, 2 Leave Fullscreen, 3 Take Over master, 4 Init Ndi, 5 init Video Inputs, 6 Reset Video Inputs, 7 Reset RenderEngines

## Control a Surface with DMX-512

- Once a [DMX Routing](#) is created, VERTEX offers you the option to **control a Surface by incoming DMX-Data**
- With **104 DMX-Channels** your are able to control most parameters of a Surface

### Settings

- Select your Surface and go to Inspector



- Select the DMX Tab into Inspector and do your settings for start address and universe
- Don't forget to activate the DMX Input for this System

DMX Channelmap for a Surface

DMX Address	Parameter Name	Type	Default Value	Note		
1	opacity coarse	16bit	0	16bit opacity 1 mode for crossfade wipe modetexture		
2	opacity fine	1				
3	Position	xpos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = - 16.000 pixel	
4		xpos fine				
5		ypos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = - 16.000 pixel	
6		ypos fine				
7		zpos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = - 16.000 pixel	
8		zpos fine				
9		Rotation	xrot coarse	216bit	32768	mode crosse fade from absolut to continous value
10			xrot fine			128 = no motion 255=fastForward 0=fastReverse
11	yrot coarse		216bit	32768		
12	yrot fine					
13	zrot coarse		216bit	32768		
14	zrot fine					
15	Scale	xscale	216bit	32768		
16		xscale fine				
17		yscale	216bit	32768		

18		yscal fine				
19		zscal e	216bit	32768		
20		zscal efine				
21		Geo metr y Fold er	1	0		
22		Geo metr y File	1	0		
23		Blen d Fold er	1			
24		Blen d File	1			
25		Col Tem p Inten sity	216bit	0		
26		fine				
27		Col Tem p Valu e	216bit	32768		
28		fine				
29	Color Correcti on	Hue	216bit	32768		
30		fine				
31		Satur ation	216bit	32768		
32		fine				
33		Contr ast	216bit	32768		
34		fine				
35		Bright ness	216bit	32768		
36		fine				

37	Red Balance	216bit	32768		
38	fine				
39	Green Balance	216bit	32768		
40	fine				
41	Blue Balance	216bit	32768		
42	fine				
43	Levels Min	216bit	0		
44	fine				
45	Levels Value	216bit	32768		
46	fine				
47	Levels Max	216bit	65535		
48	fine				
49	View Point OffsetX	216bit	32768		
50	fine				
51	View Point OffsetY	216bit	32768		
52	fine				
53	View Point OriginX	216bit	32768		
54	fine				
55	View Point OriginY	216bit	32768		
56	fine				
57	View Point	216bit	32768		

	OriginZ				
58	fine				
59	View Point TargetX	216bit	32768		
60	fine				
61	View Point TargetY	216bit	32768		
62	fine				
63	View Point TargetZ	216bit	32768		
64	fine				
65	Field Of View	216bit	0	Default 45°	0_65535 = 0 - 180
66	fine				
67	Roll	216bit	32768		
68	fine				
69	Aspect Ratio	216bit	32768		
70	fine				
71	View Point Mode	1			
72	View Point Ctrl	1			
73	Edge Blending	Left	216bit	0	
74	fine				
75	Left Start	216bit	0		
76	fine				
77	Left Gamma	216bit	0		
78	fine				

79	Left Lumi nanc e	216bit	32768		
80	fine				
81	Right	216bit	0		
82	fine				
83	Right Start	216bit	0		
84	fine				
85	Right Gam ma	216bit	0		
86	fine				
87	Right Lumi nanc e	216bit	32768		
88	fine				
89	Top	216bit	0		
90	fine				
91	Top Start	216bit	0		
92	fine				
93	Top Gam ma	216bit	0		
94	fine				
95	Top Lumi nanc e	216bit	32768		
96	fine				
97	Botto m	216bit	0		
98	fine				
99	Botto m Start	216bit	0		
100	fine				
101	Botto m Gam ma	216bit	0		
102	fine				
103	Botto m	216bit	32768		

104	Lumi nanc e					
	fine					

## Control a Console Layer with DMX-512

- Console Layers gives you the option to control video like a lighting fixture including Iris and shaper
- A Console Layer is **basically independent from a Sequence or a Playlist**. It is only assigned to a Canvas
- with **192 DMX Channels for each Console Layer** you are able to control a bunch of fine adjustments, effects, iris, shaper
- Use **Console Layer as ClipContainer** in your show to **temporary overwrite the global values**

### Channel Map

- Each Console Layer **has 192 DMX-Channels**.
- The channel map could be used as blueprint for a lighting desk fixture/a library fixture

DMX Channel Map for a Console Layer

DMX Address	Parameter Name	Type	Default Value	Note
1	opacity coarse	16bit	0	16bit opacity 1 mode for crossfade wipe modetexture
2	opacity fine	1		
3	opacity wipe mode	1	0	0-255 where 0 no wipe/Crossfade
4	opacity wipe softness	216bit	0	
5	fine			
6	volume coarse	216bit	0	
7	volume fine			

8	Position	xpos coarse	216bit	32 76 8	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
9		xpos fine			
10		ypos coarse	216bit	32 76 8	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
11		ypos fine			
12		zpos coarse	216bit	32 76 8	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
13		zpos fine			
14	Rotation	xrot coarse	216bit	32 76 8	mode crosse fade from absolut to continous value
15		xrot fine			128 = no motion 255=fastForward 0= fastReverse
16		xrot XFade	216bit		2 abs rotation 1 mode 1 rotation speed
17		fine			
18		xrot speed	216bit		
19		fine			
20		yrot coarse	216bit	32 76 8	
21		yrot fine			
22		yrot XFade	216bit		mode crosse fade from absolut to continous value
23		fine			128 = no motion 255=fastForward 0= fastReverse
24		yrot speed	216bit		2 abs rotation 1 mode 1 rotation speed
25		fine			
26	zrot coarse	216bit	32 76 8		
27	zrot fine				
28	zrot Xfade	216bit			
29	fine				
30	zrot speed	216bit			
31	fine				
32	Scale	xscale	216bit	32 76 8	
33		xscale fine			
34		yscale	216bit	32	



			768	
35	yscale fine			
36	zscale	216bit	32768	
37	zscalefine			
38	OutputFit	1		0= 1to1 Pixel 10-19 Fill, 20-29 Fit, 30-39 Horizontal Fit, 40-49 Vertical Fit
39	Playback Modes	1		0-7 Pause InFrame, 8-15 Play Once, 16-31 Play Loop, 32-47 Reserved for Ping Pong, 48-63 Play Once Opacity Zero Pause At Current Frame, 64-79 Play Loop Opacity Zero Pause At Current Frame, 80-95 Reserved for Ping Pong Opacity Zero Pause At Current Frame, 122-134 Pause at current Frame, 248-255 Pause OutFrame
40	Playback Speed	216bit	49152	1=fast reverse 64=normalPlayReverse 128 = Pause 192=Normal Play fwd 254=fast forward
41	fine			//32769-49152 slow to normal factor 0-1.0 //49152 normal play factor 1 //49153 to 65534 normal to fast fwd 1x-8x
42	In Frame	216bit	0	65535frames from first frame in
43	fine			
44	Out Frame	216bit	0	65535frames from last frame in
45	fine			
46	Gobo MixMode	1	0	Mode to determine if gobo2 is mask or how it is multiplied over gobo1, 0-10 BlackWhite, 11-20 BwInv, 21-30 Alpha, 31-40 AlphaInv, 41-50 Overlay
47	Gobo MixLevel	216bit	0	Mix Level for Gobo Blend Mode
48	fine			
49	Gobo 1 Content Folder	1	0	
50	Content File	1	0	
51	Content UV Mode	1		Texture Wrap Mode: Default, Tile, No Tile, Mirror
52	Content UV Xoffset	216bit	32768	Left to right
53	fine			
54	Content UV Yoffset	216bit	32768	bottom to top

55		fine			
56		Content UV X Scale	216bit	32768	0 = maxTiling 65535 maxScaling
57		fine			
58		Content UV Y Scale	216bit	32768	0 = maxTiling 65535 maxScaling
59		fine			
60	Gobo 2	Mask Folder	1	0	
61		Mask File	1	0	
62		Mask Transform Mode	1		Relative or Absolute Mode - Link Sizing and position to Content
63		Mask UV Mode	1		Texture Wrap Mode:Default,Tile, No Tile, Mirror
64		Mask UV Xoffset	216bit	32768	Left to right
65		fine			
66		Mask UV Yoffset	216bit	32768	bottom to top
67		fine			
68		Mask UV X Scale	216bit	32768	0 = maxTiling 65535 maxScaling
69		fine			
70		Mask UV Y Scale	216bit	32768	0 = maxTiling 65535 maxScaling
71		fine			
72		Geometry Folder	1		
73		Geometry File	1		
74		res	1		reserved for Geometry playmode
75		res	216bit		reserved for Geometry playspeed
76		res fine			
77		res	216bit		reserved for Geometry InFrame
78		res fine			
79		res	216bit		reserved for Geometry OutFrame
80		res fine			
81	Iris	Iris Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
82		Iris Size	216bit	65535	Default 65535

83		fine			
84		Iris Softness	216bit	3300	Default 3300
85		fine			
86		Iris X	216bit	32768	Default 32768
87		fine			
88		Iris Y	216bit	32768	Default 32768
89		fine			
90	Shaper	Shaper Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
91		Shaper Softness	216bit	2000	Default 2000
92		fine			
93		SL	216bit		
94		fine			
95		SL Rot	216bit	32768	Default 32768
96		fine			
97		SR	216bit		
98		fine			
99		SR Rot	216bit	32768	Default 32768
100		fine			
101		ST	216bit		
102		fine			
103		ST Rot	216bit	32768	Default 32768
104		fine			
105		SB	216bit		
106		fine			
107		SB Rot	216bit	32768	Default 32768
108		fine			
109	Color Mix	Color Mix Mode	1		Normal, Add, Multiply, InverseMultiply, Highlights, ...
110		Color Mix Level	216bit		
111		fine			
112		Red	216bit		

113		fine			
114		Green	216bit		
115		fine			
116		Blue	216bit		
117		fine			
118		Alpha	216bit		
119		fine			
120	Color Corre ction	Hue	216bit	32 76 8	
121		fine			
122		Saturatio n	216bit	32 76 8	
123		fine			
124		Contrast	216bit	32 76 8	
125		fine			
126		Brightnes s	216bit	32 76 8	
127		fine			
128		Red Balance	216bit	32 76 8	
129		fine			
130		Green Balance	216bit	32 76 8	
131		fine			
132		Blue Blanace	216bit	32 76 8	
133		fine			
134		Levels Min	216bit	0	
135		fine			
136		Levels Value	216bit	32 76 8	
137		fine			
138		Levels Max	216bit	65 53 5	
139		fine			
140		BlendMo de	1		reserved

141	FX1	FX1 Select	1			
142		FX1 Mix	216bit			
143		fine				
144		FX1 C1	216bit			
145		fine				
146		FX1 C2	216bit			
147		fine				
148		FX1 C3	216bit			
149		fine				
150		FX1 C4	216bit			
151		fine				
152		FX1 C5	216bit			
153		fine				
154	FX2	FX2 Select	1			
155		FX2 Mix	216bit			
156		fine				
157		FX2 C1	216bit			
158		fine				
159		FX2 C2	216bit			
160		fine				
161		FX2 C3	216bit			
162		fine				
163		FX2 C4	216bit			
164		fine				
165		FX2 C5	216bit			
166		fine				
167	FX3	FX3 Select	1			
168		FX3 Mix				
169		fine	216bit			
170		FX3 C1	216bit			
171		fine				
172		FX3 C2	216bit			
173		fine				
174		FX3 C3	216bit			
175		fine				
176		FX3 C4	216bit			
177		fine				
178		FX3 C5	216bit			
179		fine				
180	FX4	FX4 Select	1			
181		FX4 Mix	216bit			
182		fine				
183		FX4 C1	216bit			

184	fine				
185	FX4 C2	216bit			
186	fine				
187	FX4 C3	216bit			
188	fine				
189	FX4 C4	216bit			
190	fine				
191	FX4 C5	216bit			
192	fine				

**i** **ChamSys**  
 For ChamSys consoles, the maximum number of DMX channels of a library fixture is limited. Therefore a VERTEX Console Layer is splitted into 2 different ChamSys fixtures. Please see the ChamSys channel map below.  
 When using Console Layers with ChamSys, please switch the DMX mode of a Console Layer (Inspector Tab "DMX", Property "DMX Mode") from "Standard" to "ChamSys".

DMX-Channel Map for a Console Layer and ChamSys Consoles

58 DM X Address	Parameter Name	Type	Default Value	Note	
Fixture 1 A	1 1	opacity coarse	116bit	0	16bit opacity 1 mode for crossfade wipe modetexture
	2	opacity fine	1		
	1 3	opacity wipe mode	1	0	0-255 where 0 no wipe/Crossfade
	1 4	opacity wipe softness	216bit	0	
	5	fine			
	1 6	volume coarse	216bit	0	
	7	volume fine			
	1 8	Position coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel
	9	xpos fine			

1	10	ypos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel	
	11	ypos fine				
	12	zpos coarse	216bit	32768	32768 = 0 65535 = + 16.000 pixel 0 = -16.000 pixel	
	13	zpos fine				
	14	Rotation	xrot coarse	216bit	32768	mode crosse fade from absolut to continous value
	15		xrot fine			128 = no motion 255=fastForward 0=fastReverse
	16		yrot coarse	216bit	32768	
	17		yrot fine			
	18		zrot coarse	216bit	32768	
	19		zrot fine			
	20	Scale	xscale	216bit	32768	
	21		xscale fine			
	22		yscale	216bit	32768	
	23		yscale fine			
	24		zscale	216bit	32768	
	25		zscalefine			
	26		OutputFit	1		0= 1to1 Pixel 10-19 Fill, 20-29 Fit, 30-39 Horizontal Fit, 40-49 Vertical Fit
	27	Playback	PlayModes	1		0-7 Pause InFrame, 8-15 Play Once, 16-31 Play Loop, 32-47 Reserved for Ping Pong, 48-63 Play Once Opacity Zero Pause At Current Frame, 64-79 Play Loop Opacity Zero Pause At Current Frame, 80-95 Reserved for Ping Pong Opacity Zero Pause At Current Frame, 122-134 Pause at current Frame, 248-255 Pause OutFrame
	28		PlaybackSpeed	216bit	49152	1=fast reverse 64=normalPlayReverse 128 = Pause 192=Normal Play fwd 254=fast forward
	29		fine			//32769-49152 slow to normal factor 0-1.0 //49152 normal play factor 1 //49153 to 65534 normal to fast fwd 1x-8x
	30		In Frame	216bit	0	65535frames from first frame in
	31		fine			

1 32		Out Frame	216bit	0	65535frames from last frame in
		fine			
1 33					
1 34		Gobo MixMode	1	0	Mode to determine if gobo2 is mask or how it is multiplied over gobo1, 0-10 BlackWhite, 11-20 BwInv,21 -30 Alpha,31-40 AlphaInv,41-50 Overlay
1 35		Gobo MixLevel	216bit	0	Mix Level for Gobo Blend Mode
		fine			
1 36					
1 37	Gobo 1	Content Folder	1	0	
1 38		Content File	1	0	
1 39		Content UV Mode	1		Texture Wrap Mode: Default, Tile, No Tile, Mirror
1 40		Content UV Xoffset	216bit	32768	Left to right
		fine			
1 41					
1 42		Content UV Yoffset	216bit	32768	bottom to top
		fine			
1 43					
1 44		Content UV X Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
		fine			
1 45					
1 46		Content UV Y Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
		fine			
1 47					
1 48	Gobo 2	Mask Folder	1	0	
1 49		Mask File	1	0	
1 50		Mask Transform Mode	1		Relative or Absolute Mode - Link Sizing and position to Content
1 51		Mask UV Mode	1		Texture Wrap Mode:Default,Tile, No Tile, Mirror
1 52		Mask UV Xoffset	216bit	32768	Left to right
		fine			
1 53					
1 54		Mask UV Yoffset	216bit	32768	bottom to top



55		fine			
1 56		Mask UV X Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
57		fine			
1 58		Mask UV Y Scale	216bit	32768	32768 =0 0 = maxTiling 65535 maxScaling
59		fine			
1 60		Geometry Folder	1		
1 61		Geometry File	1		
1 62	Iris	Iris Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
1 63		Iris Size	216bit	65535	Default 65535
64		fine			
1 65		Iris Softness	216bit	3300	Default 3300
66		fine			
1 67		Iris X	216bit	32768	Default 32768
68		fine			
1 69		Iris Y	216bit	32768	Default 32768
70		fine			
1 71	Shaper	Shaper Mode	1		Black Black Inv, Alpha, Alpha Inv, White, White inv
1 72		Shaper Softness	216bit	2000	Default 2000
73		fine			
1 74		SL	216bit		
75		fine			
1 76		SL Rot	216bit	32768	Default 32768
77		fine			
1 78		SR	216bit		
79		fine			
1 80		SR Rot	216bit	32768	Default 32768
81		fine			
1 82		ST	216bit		
83		fine			
1 84		ST Rot	216bit	32768	Default 32768
85		fine			
1 86		SB	216bit		
87		fine			

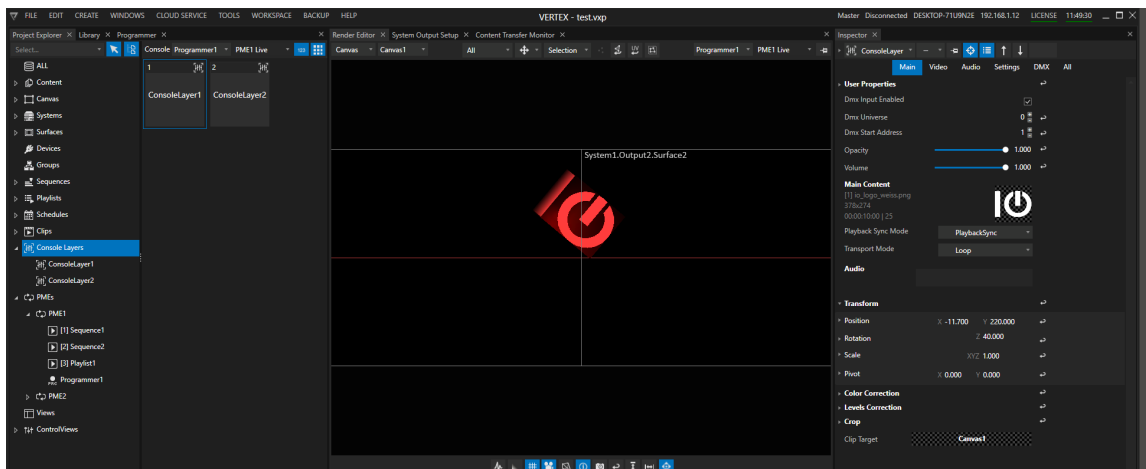
	1	88	SB Rot	216bit	32768	Default 32768
		89	fine			
	1	90	Color Mix	1		Normal, Add, Multiply, InverseMultiply, Highlights, ...
		91	Color Mix Mode	216bit		
		92	Color Mix Level			
		93	Red	216bit		
		94	fine			
	1	95	Green	216bit		
		96	fine			
	1	97	Blue	216bit		
		98	fine			
	1	99	Alpha	216bit		
		100	fine			
Fixture 1 B	1	1	Color Correction	Hue	216bit	32768
		2	fine			
	1	3	Saturation	216bit	32768	
		4	fine			
	1	5	Contrast	216bit	32768	
		6	fine			
	1	7	Brightness	216bit	32768	
		8	fine			
	1	9	Red Balance	216bit	32768	
		10	fine			
	1	11	Green Balance	216bit	32768	
		12	fine			
	1	13	Blue Balance	216bit	32768	
		14	fine			
	1	15	Levels Min	216bit	0	
		16	fine			
	1	17	Levels Value	216bit	32768	
		18	fine			
	1	19	Levels	216bit	655	

		Max			35	
	20	fine				
	1 21	BlendM ode	1			reserved
	1 22	FX1 FX1 Select	1			
	1 23	FX1 Mix	216bit			
	24	fine				
	1 25	FX1 C1	216bit			
	26	fine				
	1 27	FX1 C2	216bit			
	28	fine				
	1 29	FX1 C3	216bit			
	30	fine				
	1 31	FX1 C4	216bit			
	32	fine				
	1 33	FX1 C5	216bit			
	34	fine				
	1 35	FX2 FX2 Select	1			
	1 36	FX2 Mix	216bit			
	37	fine				
	1 38	FX2 C1	216bit			
	39	fine				
	1 40	FX2 C2	216bit			
	41	fine				
	1 42	FX2 C3	216bit			
	43	fine				
	1 44	FX2 C4	216bit			
	45	fine				
	1 46	FX2 C5	216bit			
	47	fine				
	1 48	FX3 FX3 Select	1			
	1 49	FX3 Mix				
	50	fine	216bit			
	1 51	FX3 C1	216bit			
	52	fine				
	1 53	FX3 C2	216bit			
	54	fine				
	1 55	FX3 C3	216bit			
	56	fine				
	1 57	FX3 C4	216bit			
	58	fine				
	1 59	FX3 C5	216bit			
	60	fine				
	1 61	FX4 FX4	1			

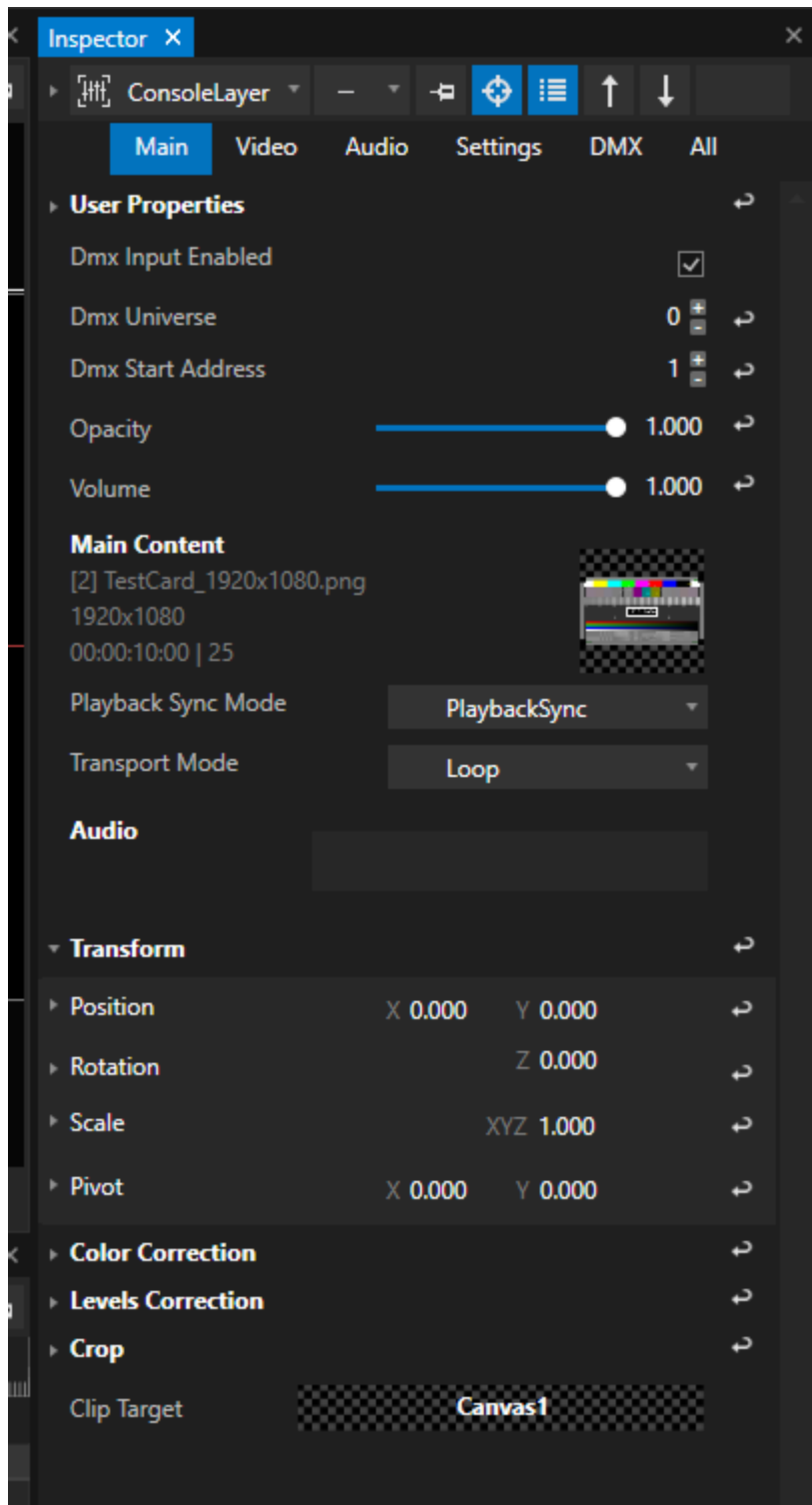
		Select			
	1	62	FX4 Mix	216bit	
		63	fine		
	1	64	FX4 C1	216bit	
		65	fine		
	1	66	FX4 C2	216bit	
		67	fine		
	1	68	FX4 C3	216bit	
		69	fine		
	1	70	FX4 C4	216bit	
		71	fine		
	1	72	FX4 C5	216bit	
		73	fine		

39

## Working with a Console Layer



- Use the "Create" tab of the Main Menu to create a new Console Layer
- Right-Click on the ConsoleLayer section into Project Explorer and use the Context Menu the create a new one



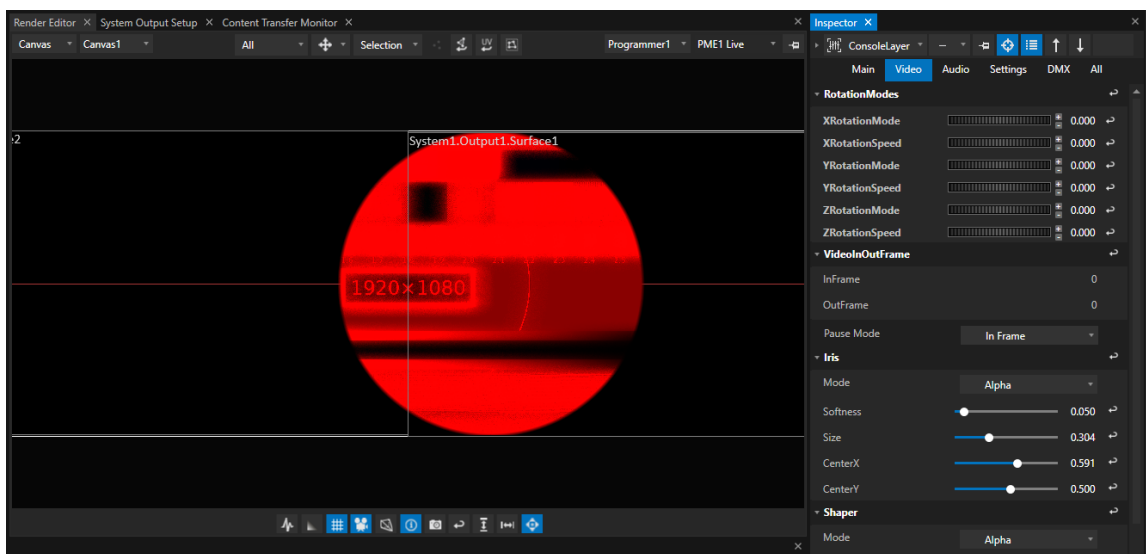
- Select Console Layer into **Inspector** and do **initial settings**:
- Set the **target Canvas** on which the Content should be rendered
- Set the **DMX universe** and **start address** if the console layer should be controlled by a lighting desk.

Please do a **DMX-Routing** first. Check the **Channel Map** for a Console Layer in detail and learn more about which DMX-512 channel controls which parameter

**!** **Unique start address for each Console Layer**  
 please keep in mind to change the start addresses for every console layer.  
 To be control separately, each console layer must have it's unique DMX start address

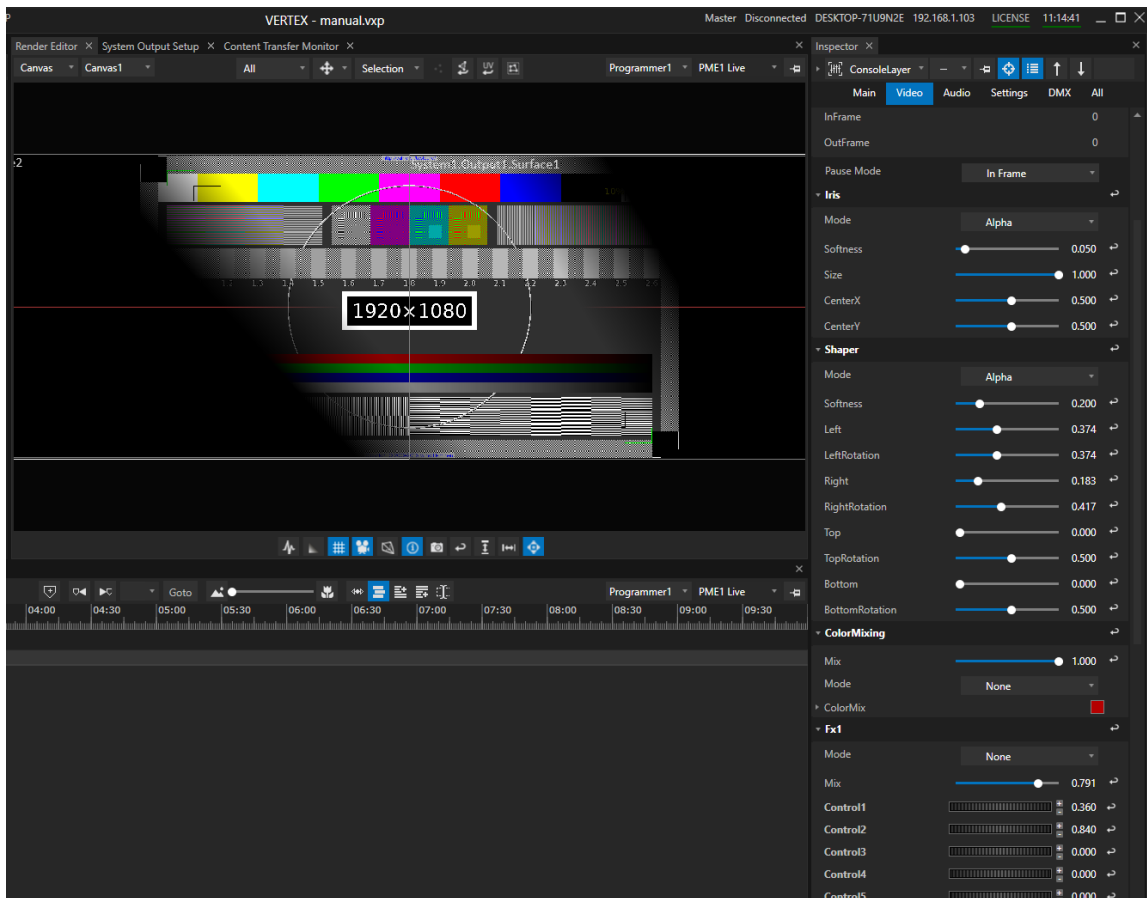
- If you want to use DMX-Values to select Content from Project Explorer like a Gobo, set **DMX File and Folder IDs** for your Content.

**i** **Channel Map: Gobo 1 and Gobo 2**  
 Gobo 1 in Channel Map is Property "Main Content" (Video tab in the Inspector)  
 Gobo 2 in Channel Map is Property "Mask" ( Video tab in the Inspector)



Iris and Color Mixing on a Example Content

- Control Console Layer by external DMX or use the Inspector to do settings for all parameters



Shaper for a Test pattern Content

## DMX Folder and File ID

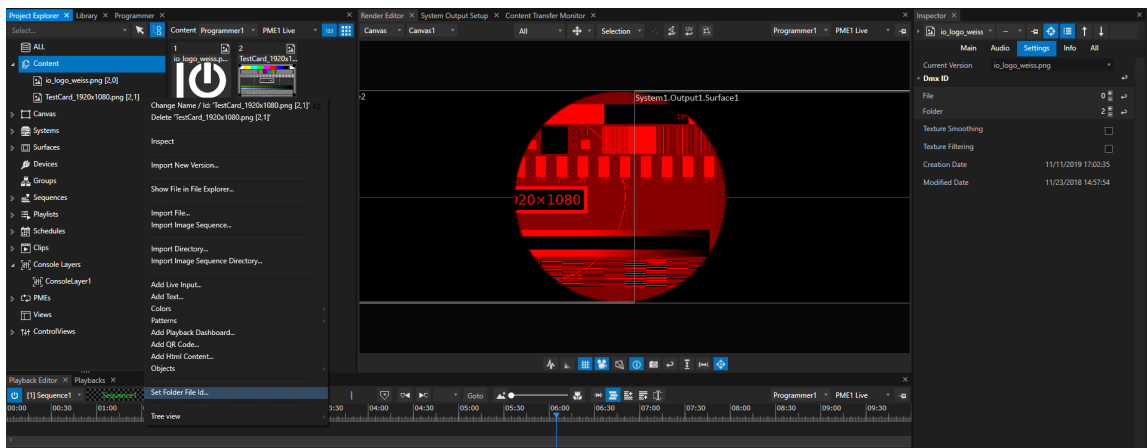
- Each Console Layer is able to host Video Content and an Content Item as a Mask.
- With the concept of folder and file IDs you are able to select your content based on a DMX Value between 0 and 255.
- Each Console Layer has a DMX Channel/Address for Content Folder ID and Content File ID (Channels 49/50) and two Channels/Addresses for Mask Content Folder and Mask Content File (Channels 60/61). Please also compare with the full [channel map of a console Layer](#)

### Example:

- Your Console Layer into VERTEX has DMX Start Address 1 (For universe routing, please read topic [DMX-Routing](#) before)
- Your lighting desk has a library element for a VERTEX Console Layer, also on Start Address 1
- you want to select a Content Item from the VERTEX Project Explorer with File ID 5 and Folder ID 2 as Video Content for this Console Layer

- you have to set DMX-channel 49 of your lighting desk to value 2 and DMX channel 50 to value 5
- The content item is selected as video content for the VERTEX Console Layer

### Set File and Folder ID

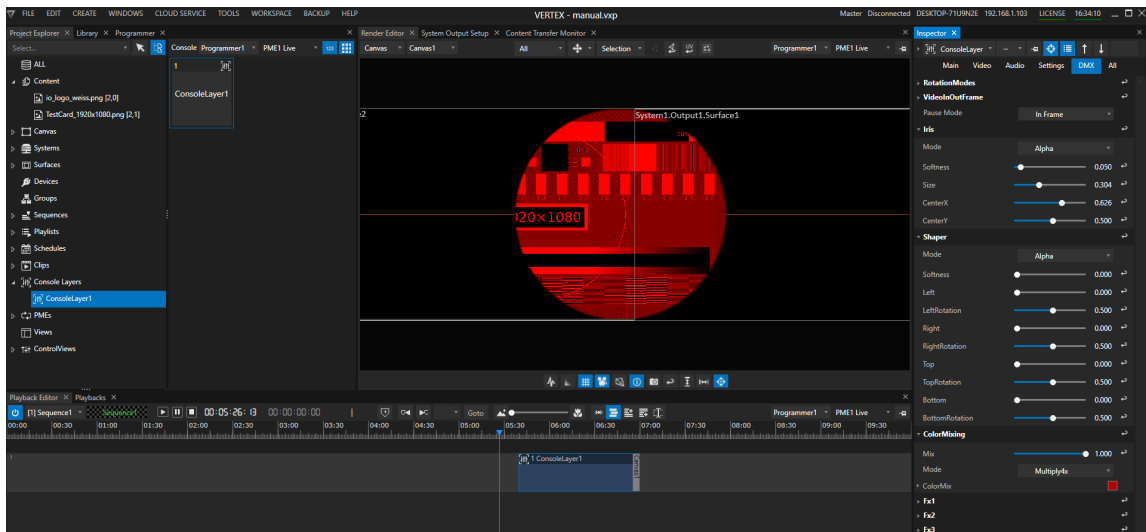


- Select a Content item into Project Explorer and go to the Inspector
- Go to the Tab "Settings" and Select "DMX ID"
- Set File and Folder ID for a Content. Each value has to be into the DMX parameter range from 0 to 255

### Clip Container for a Console Layer

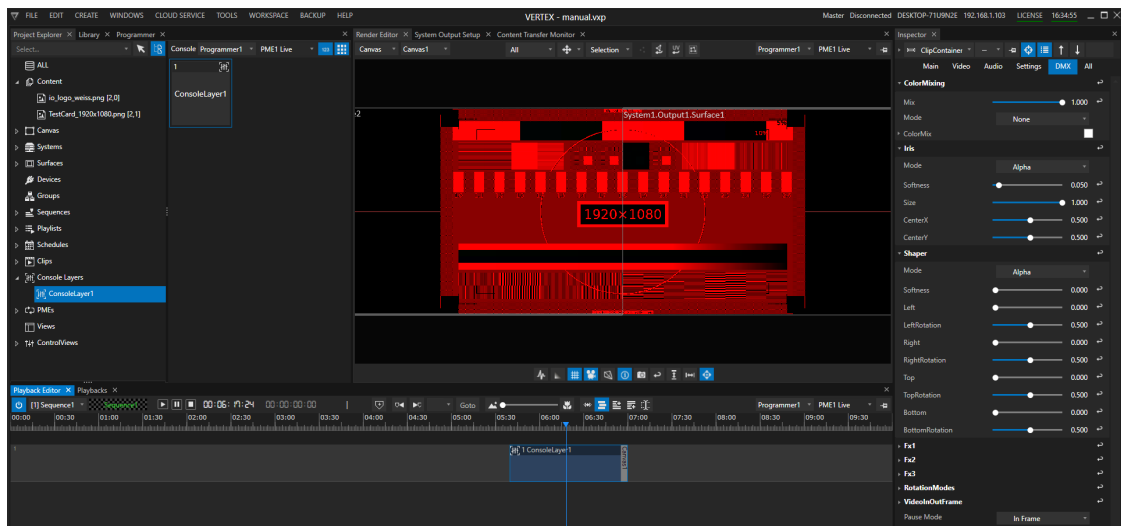
- Even for Console Layers you can create a Clip Container
- Used in a Playback, Clip Containers for a Console Layer overwrite the global value of the Console Layer





Global Setting for ConsoleLayer 1: Iris is set

- Drag a Console Layer from Project Explorer into the Playback Editor: A new Clip Container is created into your Sequence
- Assign Content to your Clip Container and set Values



Clip Container with Console Layer 1 inside overwrites the global values of Console Layer 1: in this example, all values for the iris are reset to default values.

Don't forget to also assign the same Main Content as the global Console Layer has to the Clip Container



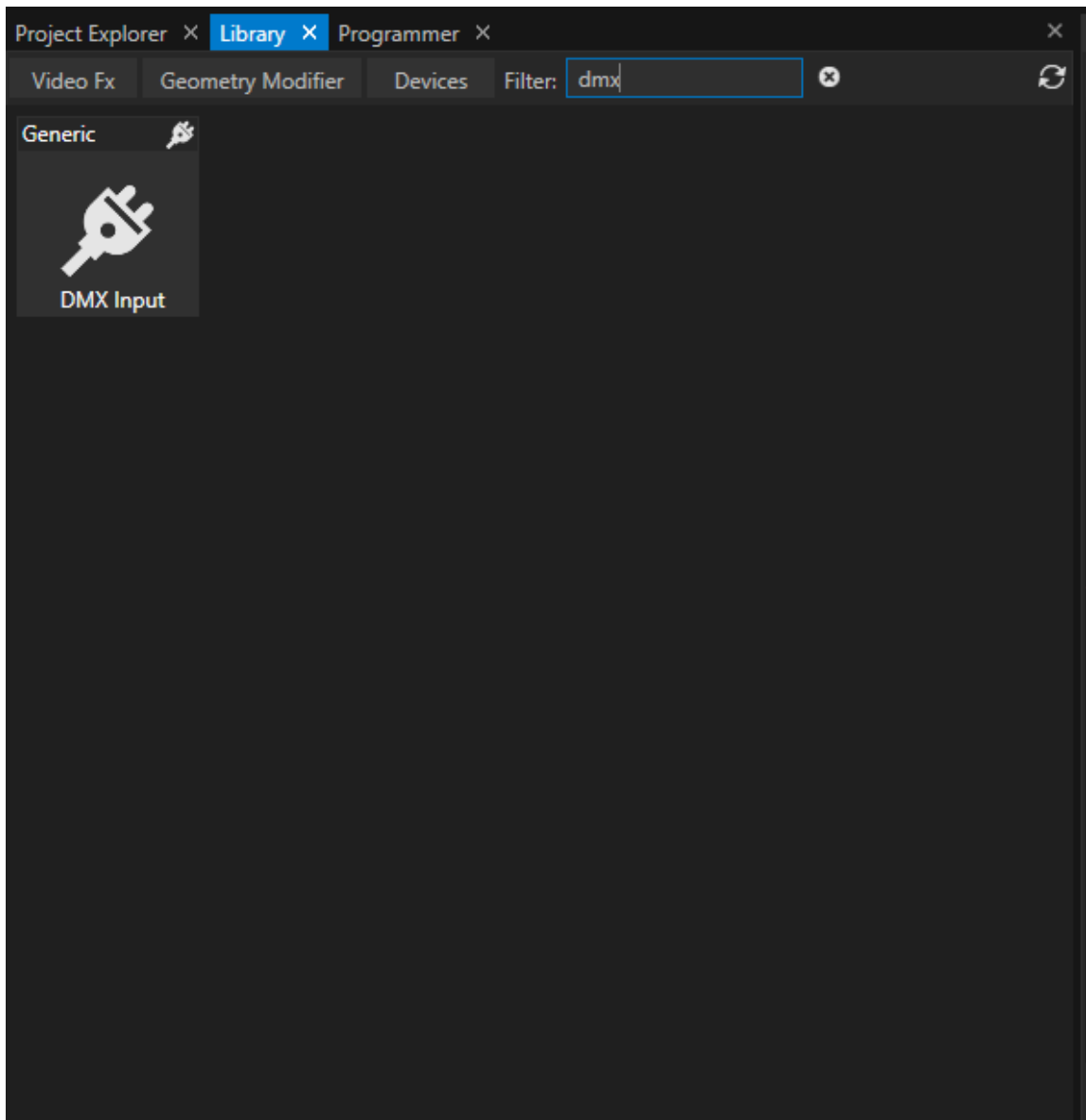
**Use case example:**

You are able to e.g. temporary take control back from a lighting desk control and to set all content to black.

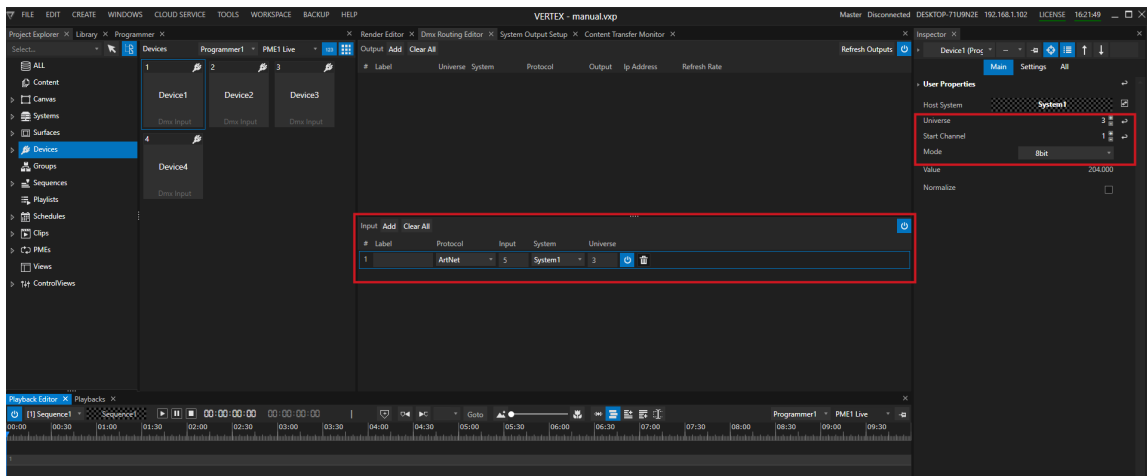
## DMX-Input Device

- Use DMX Input Devices to **trigger, wire or script interactions based on values of a DMX Channel**
- You are able to **listen on DMX Values** of a predefined DMX Start Address and Universe
- Use the **incoming DMX Values to influence other properties into VERTEX** or to trigger an action

### Add Device and do Settings



- Go to the Library Editor
- Select the tab "Devices"
- Search for "DMX Input"
- Right-Click on the Device and add one or multiple to your project



Example of a DMX Device Setting: There is DMX Input Routing. VERTEX is listening on Art-Net Universe 5. Data from this Universe is internally routed to VERTEX Universe 3. DMX Input Device 1 is set to VERTEX Universe 3, Start Channel 1.

- Go to the Device section in the [Project Explorer](#)
- Select on or multiple of your DMX Input Devices
- Set VERTEX universe number and start channel and define the DMX mode (Default 8 Bit, 16 Bit, 32 Bit...)



### Normalize Input

Enable "Normalize" in the Device settings, if you want to control or wire a property into VERTEX that has a value range from 0 to 1 (e.g. opacity).

All incoming DMX Data is normalized into the range from 0 to 1.

- Ensure, that a valid [DMX Input Routing](#) exists.

## Working with DMX Input Devices

As for every Device, there are different options to work with the incoming Data.

Here some Examples:

### 1. Scripting

The Opacity of Clip Container 1 is controlled by the incoming DMX Data from Device 1

The opacity value range is 0 to 1. The "Normalize" data option is enabled for the input Device

A [Script Command](#) for this scenario could be like this:

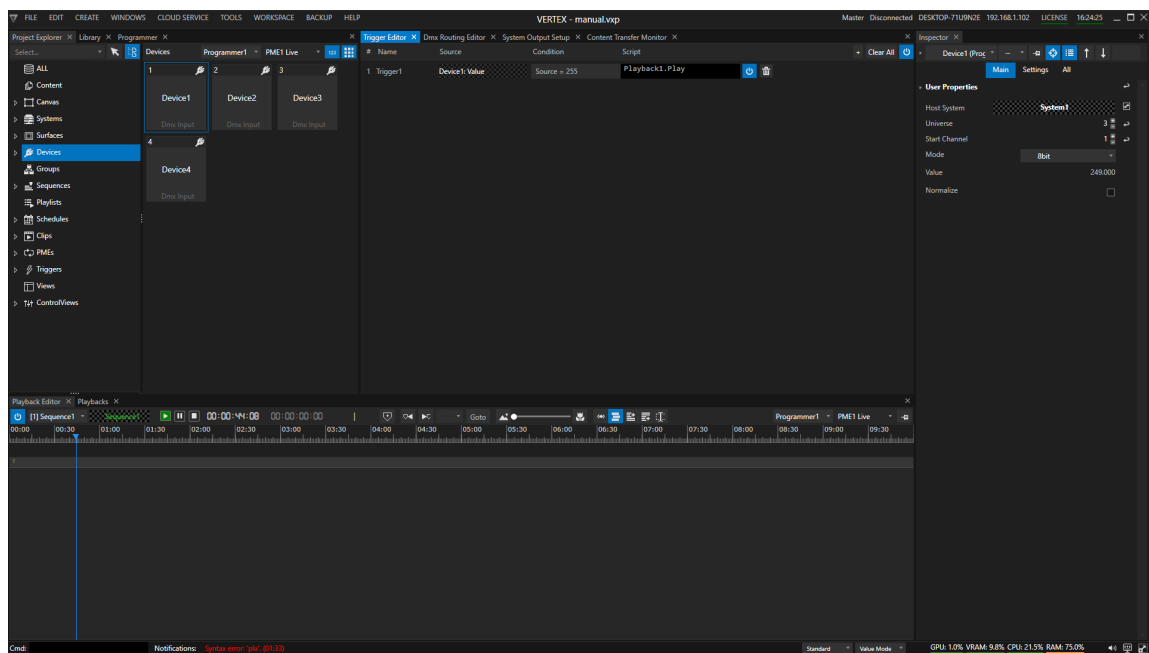
```
Sequence1.ClipContainer1.Opacity.Value = Device1.Settings.Value.Value
```

### 2. Trigger

With help of the [Trigger Editor](#) you can easily build Triggers. An defined action is executed if a condition for a source is reached

Of course this also works with DMX Input Devices

- Open the [Trigger Editor](#)
- Drag the e.g. Value property from Inspector to the Source field
- Enter a [Condition](#)
- Enter a [Script](#) that should be executed



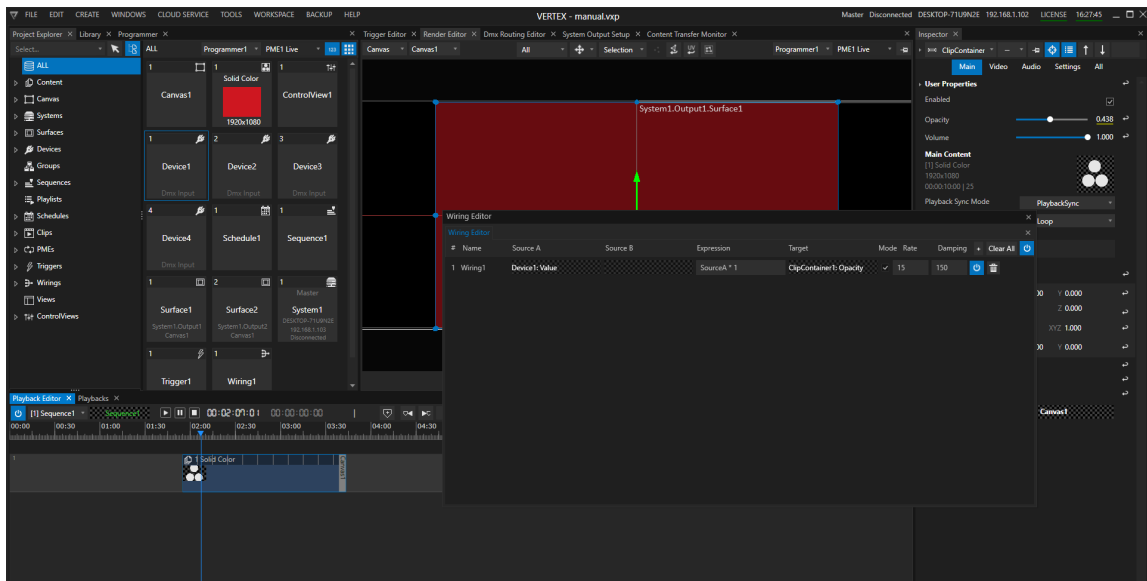
Example for the usage of a DMX Input Device:  
If DMX-Value of Universe 3/Startchannel 1 is 255, Playback 1 is started

### 3. Wiring

[Wiring](#) gives you the option to connect Values of different properties.

Of course this also works for DMX-Input Devices

- Open the [Wiring Editor](#)
- Drag a DMX Device Property from Inspector to e.g. the Source field
- Drag another property into target field
- Add an expression



Example of a Wiring  
The Value of the DMX Input Device 1 ( Normalized) is wired to the Opacity of Clip Container 1.  
The Opacity now is controlled by a DMX Channel.

## 6.7.4 DMX-Output

- With **different types of DMX Output Devices** you are able to send out DMX Data
- Integrate DMX Devices to your show, **program lighting scenarios based on keyframes** or work with them in **programmer mode**
- Create a **group of Devices** and control them together with only one Clip Container

## Workflow

### Add from Library

- Open the **Library Editor**
- Select the **Devices** Filter there
- **Search for your DMX Device**



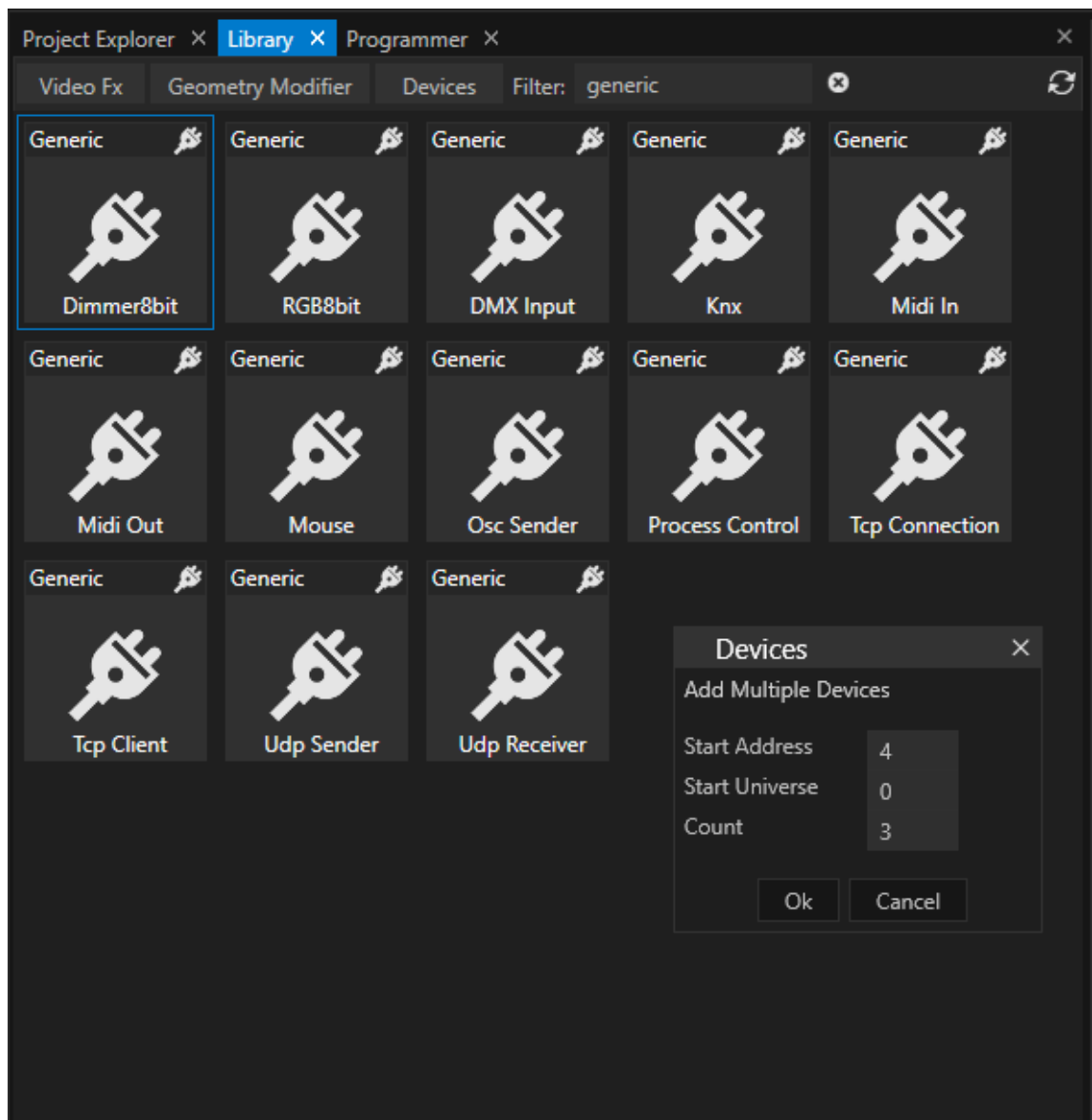
#### **DMX Devices and VERTEX Library**

Depending on the VERTEX Assembly version, the available number of different DMX Devices could change. Basically, VERTEX is shipped with Generic standard devices: a 8 Bit Dimmer and and 8 Bit RGB Device. If you require other DMX Devices for your project, just write us an email to [support@ioversal.com](mailto:support@ioversal.com) with type and in the most ideal case directly with the DMX

table :-)

- **Right-Click** on the Device you want to add
- Select "**add to project**" or "**add multiple**"
- If you only have **added one device**: select the device into Inspector to set DMX Universe and Start address
- If you have select "**added Multiple**", a dialog window opens. Just add the **start address and universe** there and the number of devices you want to add.

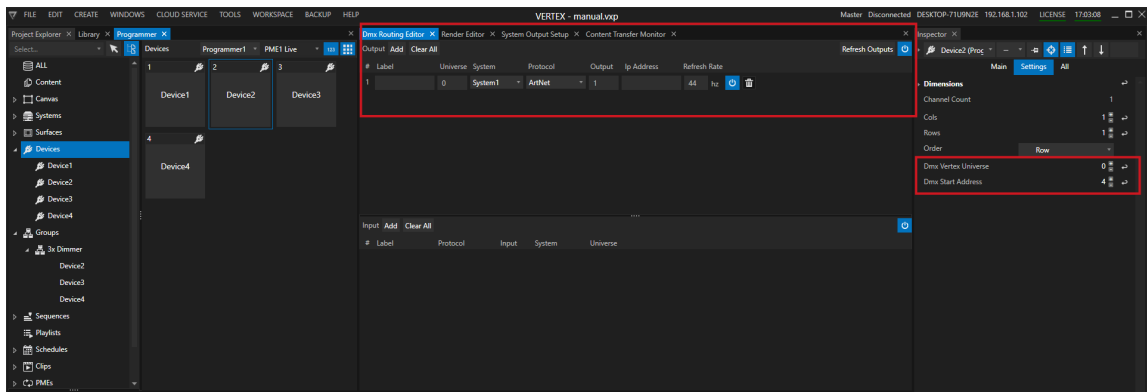
The first device will start with start address and universe you have set. For all other devices, start address and universe are automatically set by VERTEX.



## Check DMX Routing and Start Addresses

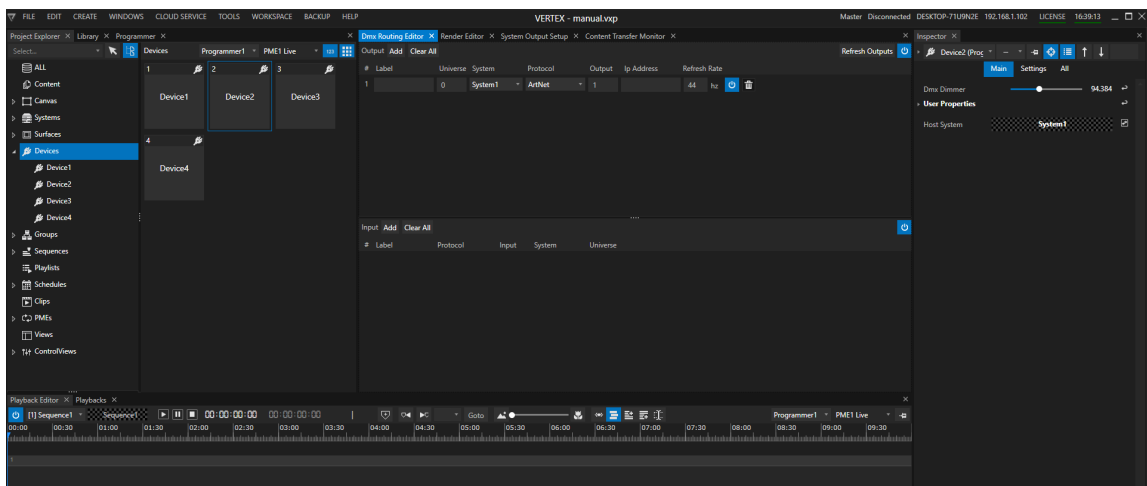
- Remember that a DMX output only will work, when a **valid [DMX routing](#)** is created. If this was not done before, please create a routing now.  
Into the [DMX Routing Editor](#) you define, which protocol is used to send DMX Data out. To send physical DMX-512, ioversal provides an USB-DMX interface, the [<%DMX\\_IO%>](#)
- Go to the Device section of the Project Explorer. select a DMX device there and set or check the **DMX Start Address** or the **Start Universe** in the Inspector.



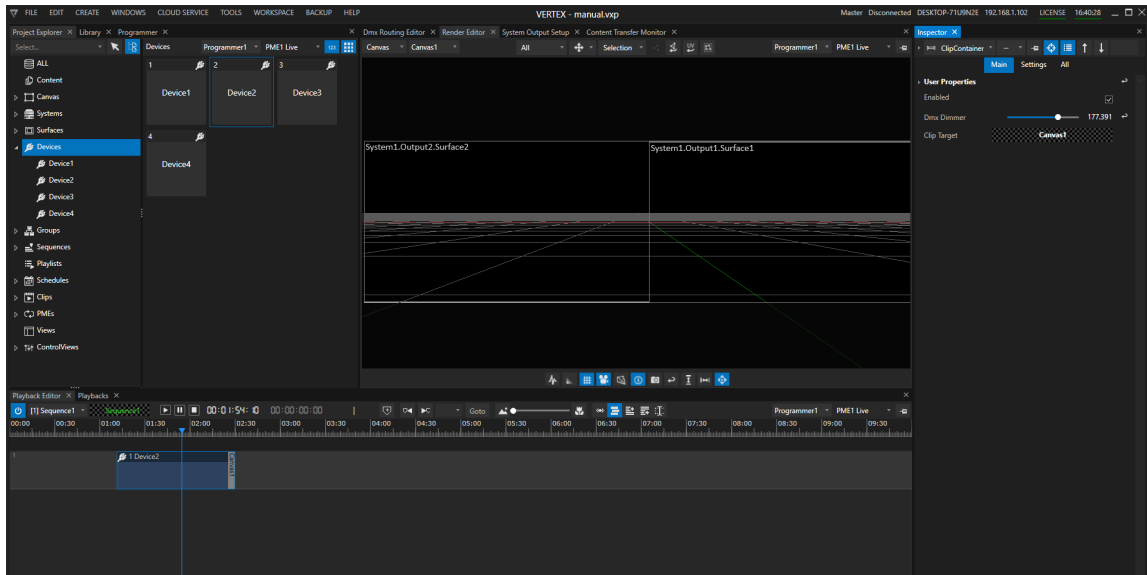


## Send Out Global Values

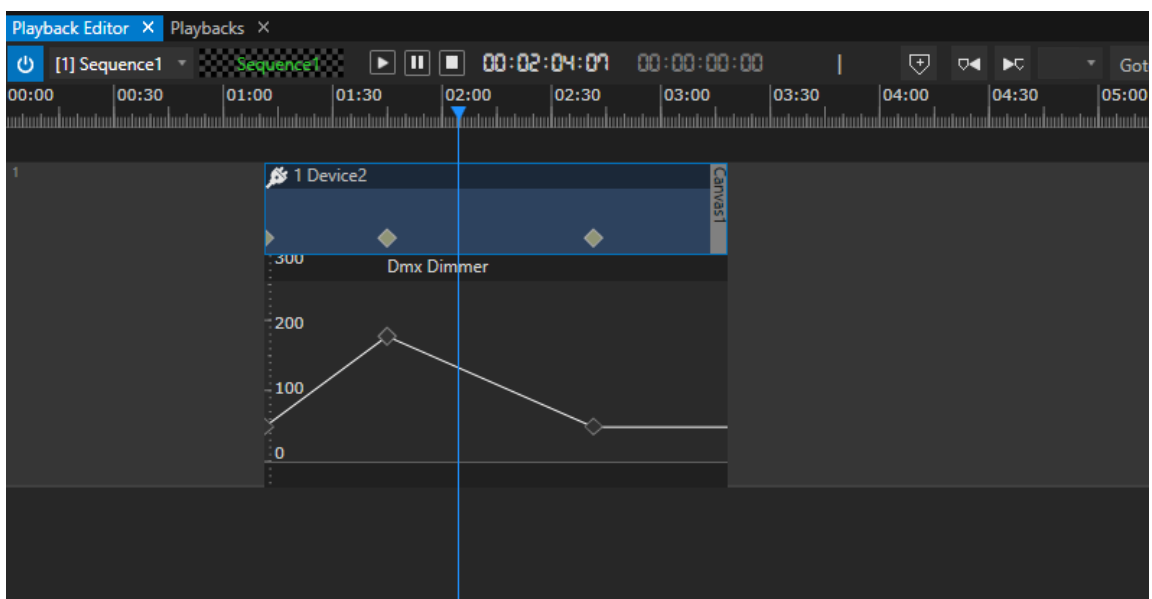
- You are able to send out a global value for your DMX device, e.g. for testing or if there generally should be a minimal value for such devices all the time.
- To set values globally, select a device directly into Project Explorer and set the DMX Value in the Inspector.
- This value is sent by vertex permanently



## Work with Clip Containers and Keyframes



- A combination of [Clip Containers](#) with [Keyframes](#) allows you to create more complex lighting scenarios, color transitions or an integration of lighting into your video show.
- Device values from a Clip Container temporary **overwrites** [global values](#) (if such are set)
- Just drag a device from Project Explorer into the playback Editor to **create a new Clip Container**
- Double click on the Clip Container to open the **Keyframe Editor** (to learn more about Keyframe animation, please read the topic [Keyframes](#))



- If you prefer a **workflow from a Lighting Desk** - you can also work in **Programmer Mode**:  
Change all values and create your scene. All changes are temporarily stored as a list in the [Programmer](#). When your scene is final, you can save the programmer list as a scene: VERTEX automatically creates the Keyframes for all changed values at the Playhead position.

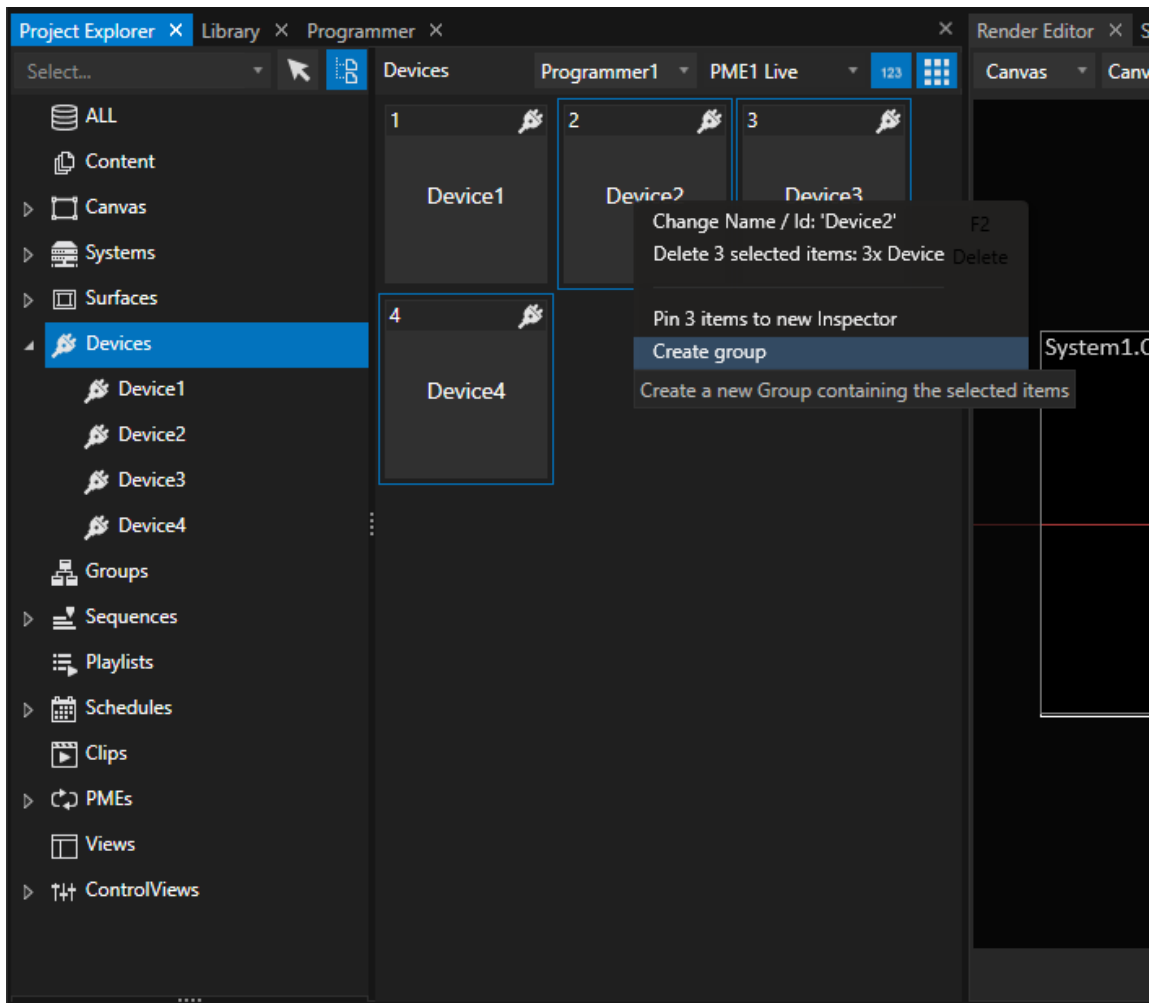
## Groups

- Manage and summarize your DMX Output Devices in Groups. Make value changes for all devices of a group

### Create a Group

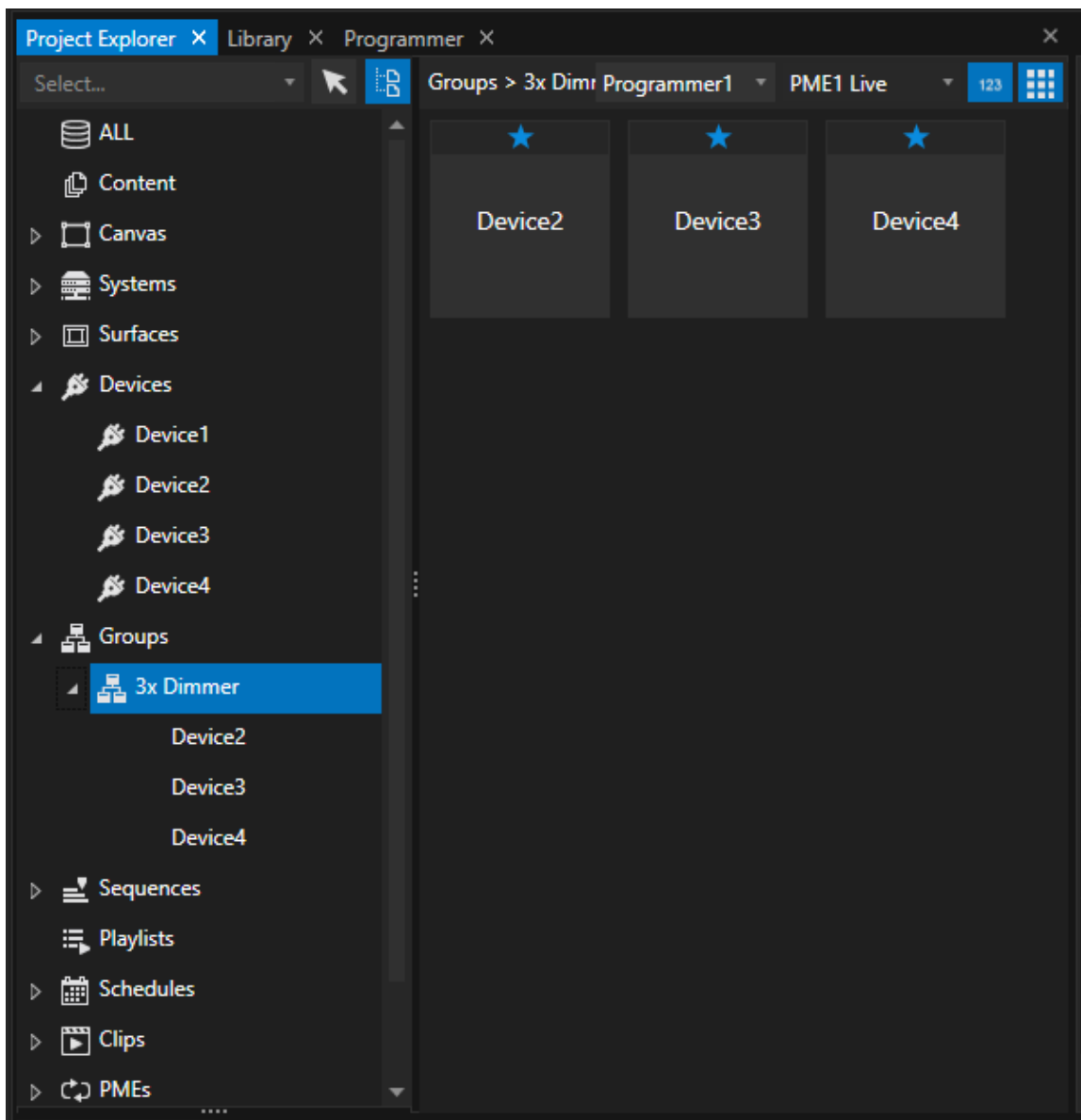
**There are 2 ways to create a group and assign Devices to it:**

- Go to the Device section into the Project Explorer
- Mark all devices that should be member of a group
- Right-Click and select "Create Group"
- A new Group that contains the selected Devices is created. You can find, select or rename the group in the "Group" section of the Project Explorer



or:

- Go to the "Groups" section in the Project Explorer
- Right-Click and open the context menu
- Choose "Add new" there
- Drag devices from Device section of the Project Explorer to this group

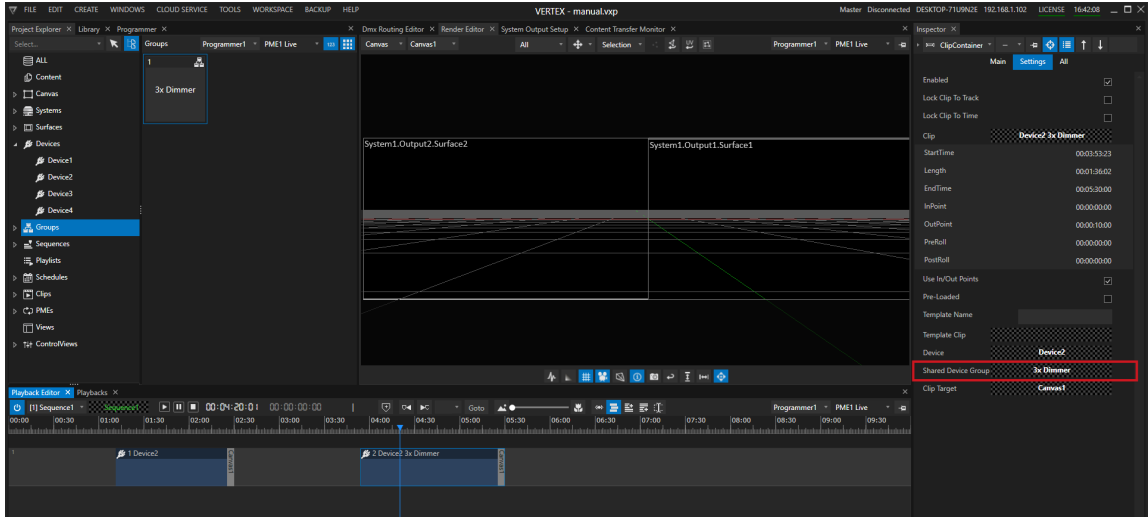


## Work with a Group of DMX-Devices

*When working with a group, you are able to set DMX Values for all group members with only one Clip Container.*

- Drag a Group from Project Explorer to the Playback Editor
- All changes you do for this Clip Container are adopted for all devices of this Group
- If values apply to a group, you can identify this with the "SharedDevice Group" Property. If there is a group set, all values of the clip container are also assigned to the group members.
- You can also transfer the settings for a DMX Device to a Group afterward by dragging a Group from Project

Explorer to the "SharedDevice Group" field of an already existing Clip Container (that already hosts a DMX Device)



## 6.8 Interaction

- VERTEX offers you **Solutions and Editors** to **make Interaction as simple and fast as possible**
- Connect Properties with help of **Wirings** or **trigger** an action based on incoming data

### Wiring Editor

[Connect Properties and do a mathematical Operation](#)

### Trigger Editor

[Execute Script Commands based Conditions, triggered by Property Values or incoming Data](#)

### Script Commands

Most items into VERTEX do have a Script Window to write and enter Scripts.

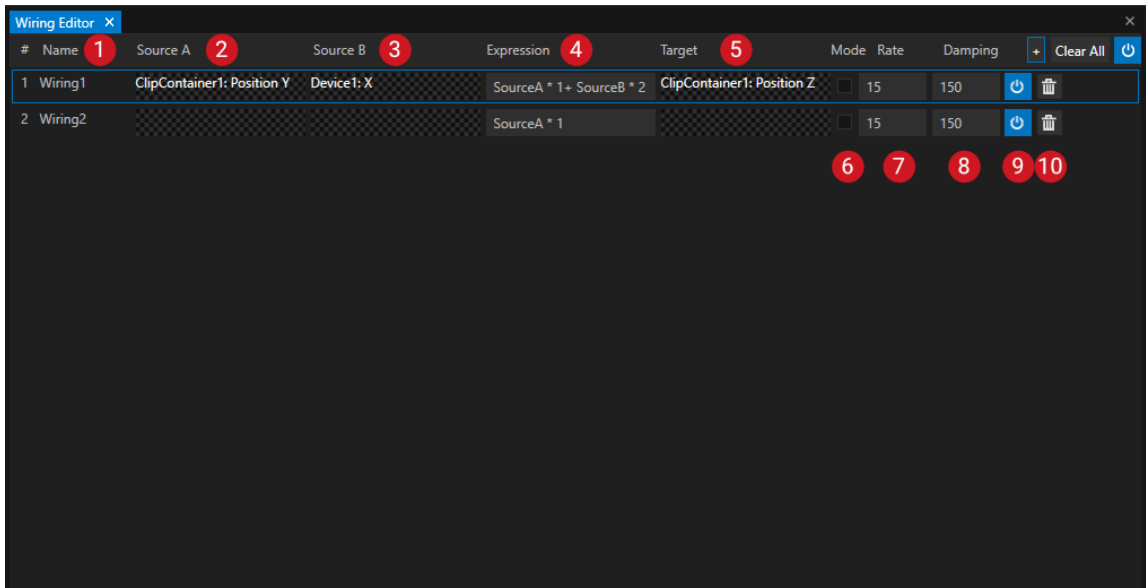
Use this Script Commands to built interaction or write Scripts that combine several Commands

[Get more information about Script Commands](#)

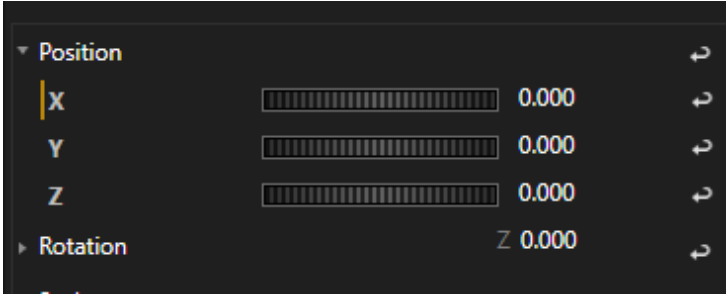
### 6.8.1 Wiring Editor

- Wiring assists you to **create simple interactive logic** and to **connect different Properties that influence each other**
- **The Wiring Editor** follows **a easy table scheme**: Define 1 or 2 Sources, enter an Expression that influence a Target Property - do expression- influence target parameter
- Sources are assigned by **drag and drop them from Inspector** - wired Properties are **underlined in yellow color** in the Inspector

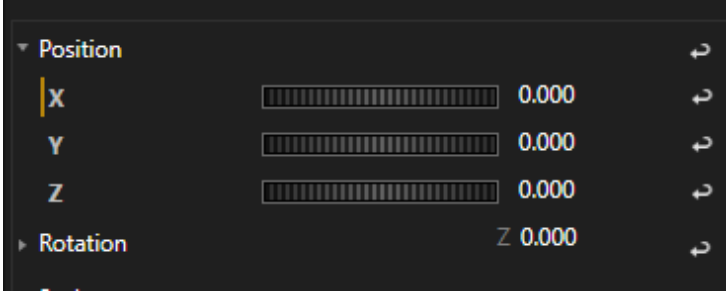
## User Interface



Example for Wiring:Y-Position of Clip Container1 is Source 1.A Mouse device that Captures the mouses x position is Source B. Both influencens the z-Position of Clip Container 1. The Expression defines how the values for the Clip Containers Z position are calculated.

1	<b>Name</b>	Enter a custom Name for your wiring
2	<b>Source 01 and Source 02</b>	<p>Add a Source by drag a Property Name from Inspector to the Source field of the Wiring Editor.</p> <p>A yellow mark shows whether if this Property could be wired or not.</p>  <p><b>Clear Sources:</b> Right-Click with your mouse into a Source field. A context menu with a "Clear" option is shown.</p>
3	<b>Expression</b>	<p>Enter a mathematical Expression.</p> <p>Use <b>Variables "SourceA" and/or "SourceB"</b> into your Expressions</p> <p>If only using 1 Source, only use 1 of them for your Expression</p>

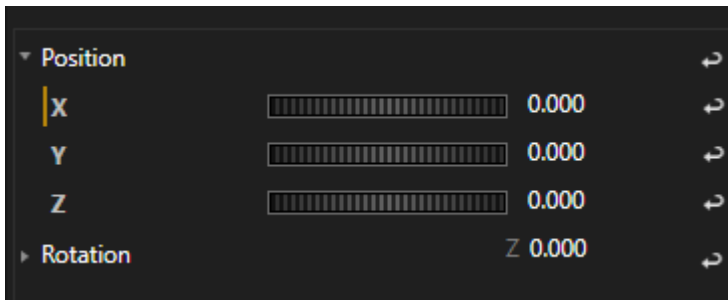


4	<b>Target</b>	<p>Add a Target by drag a Property Name from Inspector to the Target field of the Wiring Editor.</p> <p>A yellow mark shows whether if this Property could be wired or not.</p>  <p><b>Clear Targets:</b> Right-Click with your mouse into a Target field. A context menu with a "Clear" option is shown.</p>
5	<b>Mode</b>	<p><b>Default:</b> Disabled - Updates for Target Property is only done if Source Value changes</p> <p><b>When enabled,</b> the Target is permanently updated, independent from Source Value changes</p> <p><b>Disabled is recommended</b> for most cases to preserve performance: disables</p>
6	<b>Rate</b>	<p><b>Property Update Rate:</b> Value updates per seconds</p>
7	<b>Damping</b>	<p>Enter a damping for your target to avoid jitter or fast movements of your target</p> <p><b>Damping Values are set in Milliseconds</b></p>
8	<b>Activate or Mute</b>	<p>Switch Wiring on or mute your Wiring</p>
9	<b>Delete</b>	<p>Click to delete</p>

## Create a Wiring

1. **Open the Wiring Editor** from Main Menu (Main Menu -> Windows)
2. Click on the "+" Symbol to create a new Wiring

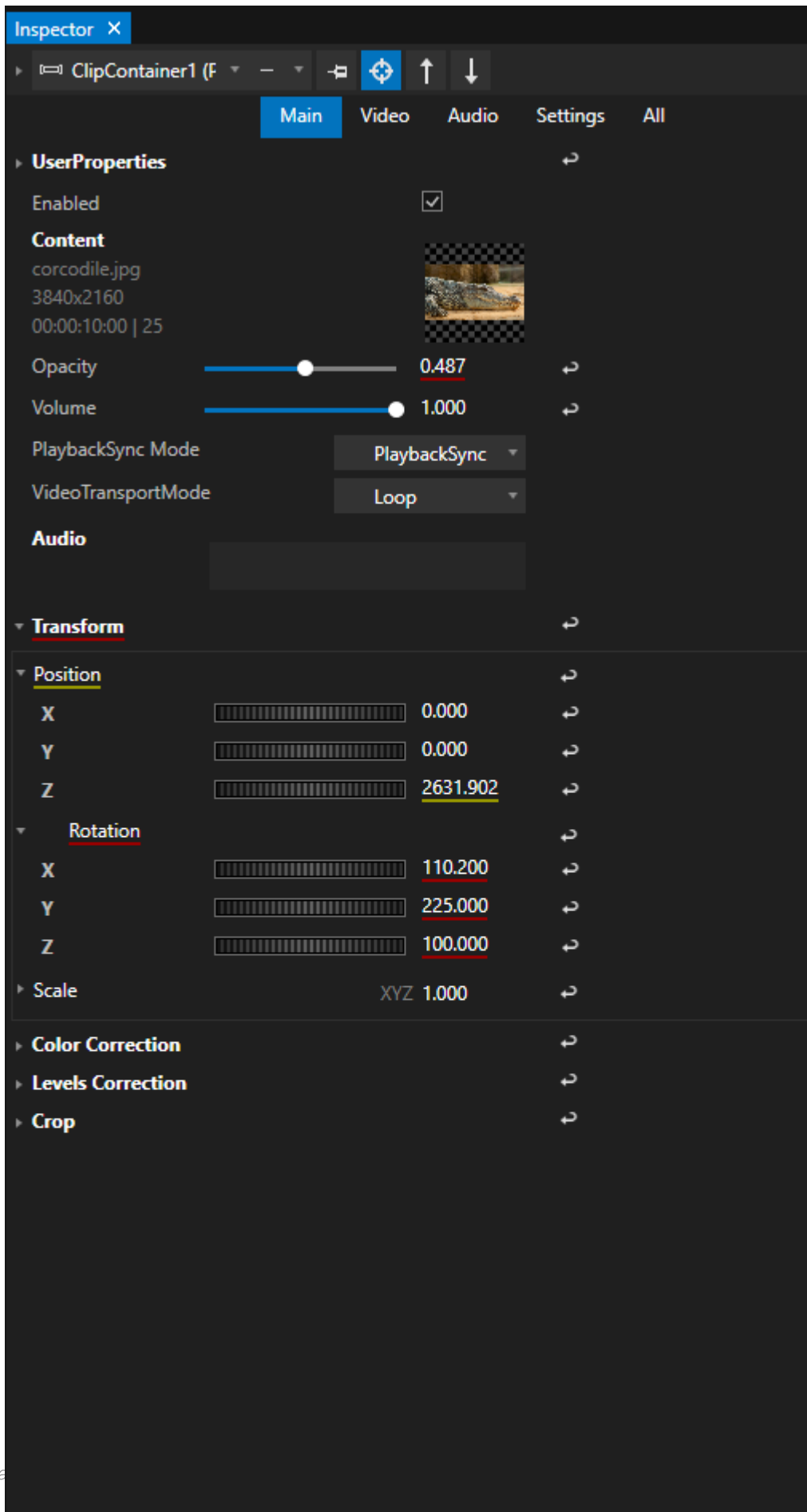
3. Select the **Item that includes your Source Property into Inspector**
4. Move with your Mouse to the Property Name into Inspector
5. When there is a **yellow mark** next to the Properties Name, **drag with your Mouse the Property Name to the Source field into Wiring Editor**



6. **Enter an Expression** - e.g. `SourceA + 1000`
7. Select a **Target Property** by **repeating Step 3 to 5** and Drag Property into the Target field of the Wiring Editor

## Yellow lines for active Target Properties into Inspector

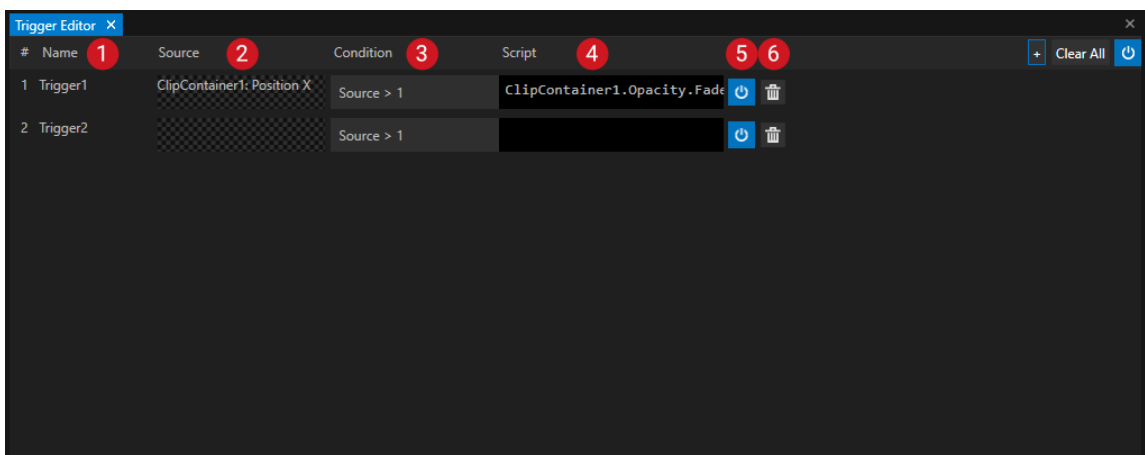
- When a Wiring is active and running, the **Target Property into Inspector is underlined in yellow Color**



## 6.8.2 Trigger Editor

- Triggering in VERTEX is an **simple way to run Script Commands based on a Condition**
- **The Trigger Editor follows a easy table scheme:** Define 1 Data Source, enter a Condition and a Script Command that should be executed when the Condition is reached
- Sources are assigned by **drag and drop them from Inspector.**

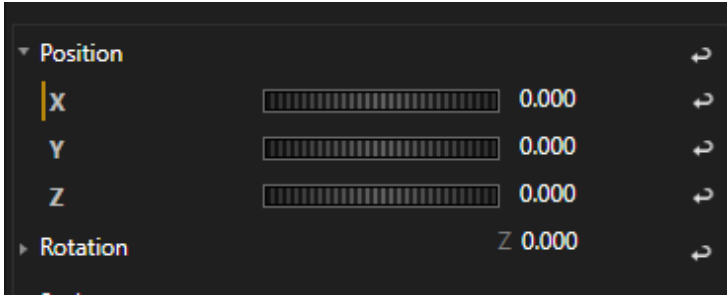
### User Interface



Example for Triggering:

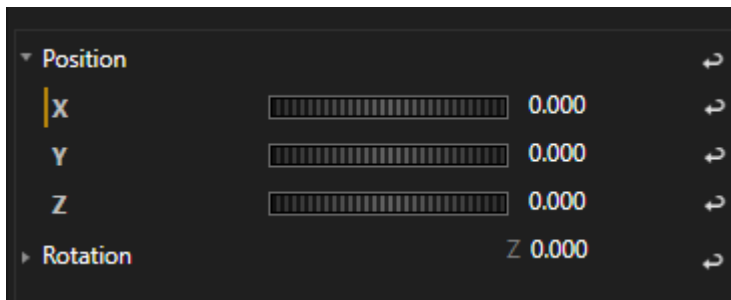
The X-Position of Clip Container 1 triggers the Opacity of this Clip Container. If The X-Position is bigger than Value 1, the Opacity is set to 0.

<b>1</b>	<b>Name</b>	Enter a custom Name for your Trigger
<b>2</b>	<b>Source</b>	Add a Source by drag a Property Name from Inspector to the Source field of the Trigger Editor.

		<p>A yellow mark shows whether if this Property could be wired or not.</p>  <p><b>Clear Sources:</b> Right-Click with your mouse into a Source field. A context menu with a "Clear" option is shown.</p>
3	<b>Condition</b>	<p>Enter a mathematical Condition . Use <b>Variable "Source"</b> into your Condition to reference the Trigger Source</p>
5	<b>Script</b>	<p>Enter a VERTEX <a href="#">Script Command</a> that should be executed when the Condition is reached Press "CTRL and Space" Keys to display a list of all available Commands and Items</p>
6	<b>Activate or Mute</b>	Switch Trigger on or mute Trigger
7	<b>Delete</b>	Click to delete a Trigger

## Create a Trigger

1. **Open the Trigger Editor** from Main Menu (Main Menu -> Windows)
2. Click on the "+" Symbol to create a new Trigger
3. Select the **Item that includes your Source Property into Inspector**
4. Move with your Mouse to the Property Name into Inspector
5. When there is a **yellow mark** next to the Properties Name, **drag with your Mouse the Property Name to the Source field into Trigger Editor**



6. **Enter an Condition** - e.g. `Source > 1000`
7. *Enter a Script Command that should be executed when the Condition is reached*  
e.g. `Playback1.Play`

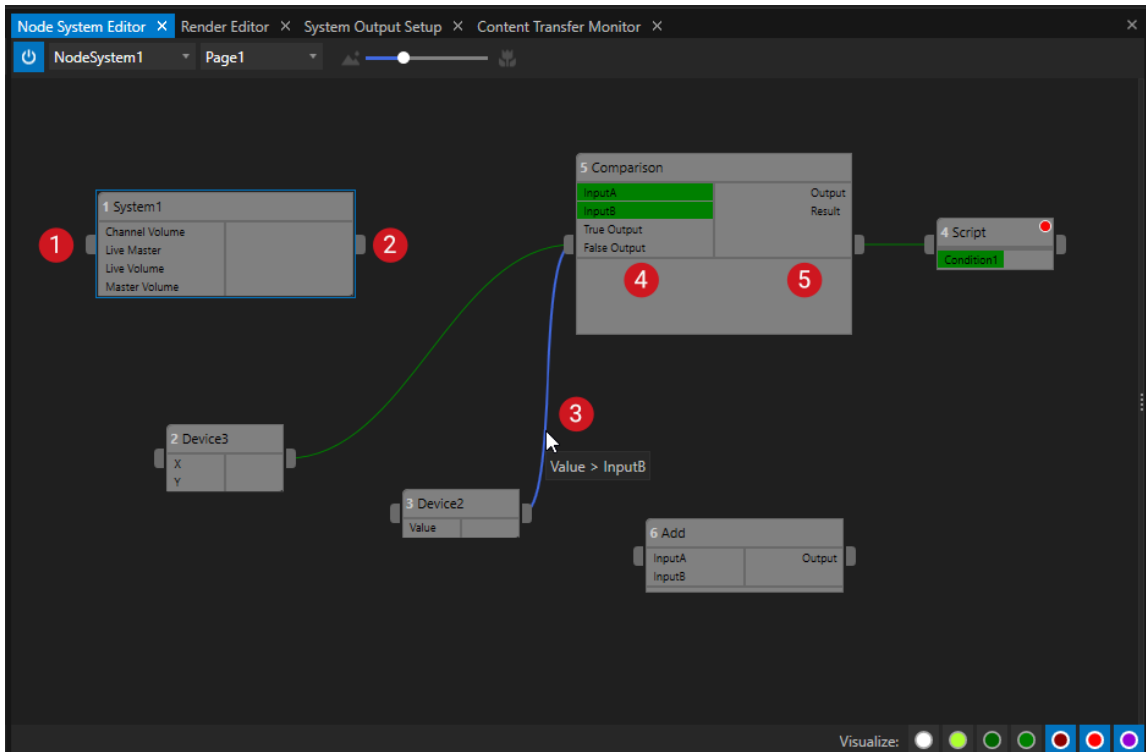
## 6.9 Node System

- Learn about the **Basics of Node programming** with a Node System in VERTEX
- A Node System is working with a **Pulse** to calculate data for each node. Learn about the different settings and options of this Pulse
- There are 2 Editors for Node programming: The **Node System Editor** and the **Node System Monitor**
- Learn about the **Node programming workflow** and what **Parameters** and **Conditions** are.
- Create a Sub-Composition from a group of Nodes and use it as **Compositing Nodes**

### Node System Basics

#### Data Flow - Input and Output

- Node Systems give you the possibility to **program complex logic** on an **easy visual way** without writing code.
- All data is **directly processed**, results are directly visible.
- All data flow and data processing is represented via **Nodes** which have a **Data Input (on the left)** and an **Data Output (on the right)**.
- You can **draw with your mouse a connection line** from an **Output** of a Node 1 to an **Input** of another Node 2. A connection line **enables the data flow** between the two nodes: Information always flows from an output of Node 1 to an input of Node 2.
- Some Nodes can manipulate or generate data. Others represent parameters from an VERTEX item (e.g. a Clip Container, a Canvas or a System). And there are Nodes that compare data or start a script command.



1	<b>Input of a Node</b>	<i>Connection for incoming data</i>
2	<b>Output of a Node</b>	<i>Connection for outgoing data</i>
3	<b>Connection Line</b>	<i>Connects an Output of a Node with an Input of another Node. Data always flows from an Output to an Input</i>
4	<b>Input Parameter</b>	<i>Data Parameter of this Node. For Nodes with only Properties and values ( and no own data manipulation or generation): there is no extra output parameter. the data of the parameters are directly on the output</i>
5	<b>Output Parameter</b>	<i>Generates own values by manipulating data applied to an input or by generating own data Nodes, where only the left column is filled, do not manipulate any data but only return values of the specified parameters.</i>



## Types of Nodes

- VERTEX offers a wide range of different Nodes.
- Some are shipped with VERTEX out of the box, some you can assemble yourself from properties of items and devices.
- Or you combine different Nodes to a Composite Node.



### **Number and type of available Nodes depends on release version**

Depending on the VERTEX release version, the number of nodes will increase step by step.

The Node System is introduced with VERTEX Release 2021 R2.

With all upcoming releases the number and also the functionality of Nodes will increase.

To give you a **first impression and overview** - nodes can **basically be classified according to the following scheme**:

### **Input (active)**

These nodes must be triggered externally to update their values and trigger a pulse if necessary, e.g. for a keyboard device or TCP Receiver device added as item from the Library

### **Input (passive)**

These Nodes provide values from external sources, e.g. a mouse input device, which are retrieved at any time (without event and "live"). They supplies current values at any time.

### **Output (active)**

These Nodes trigger external events during processing, e.g. log output or script.

An active Input is necessary to generate an external event

### **Output (passive)**

These Nodes provide current values at any time, which they determine from their inputs based on a specific implementation, e.g. mathematical Nodes or filters like add, multiply, normalize...They supplies current values at any time.

## Special Node Types

### **Property Nodes**

These Nodes are based on properties of a project item (e.g. the height of a Canvas, the position of a Clip Container, the Opacity of a Surface, the Master Volume of a System).

These Nodes are first reading the current value of a property.

Or they are modulate a Property value based on the incoming Node Input.

You are able to build own property Nodes and combine properties from different items into a single Node.

### Composite Nodes

Nodes that are a Sub-Composition of other Nodes

## Pulse(s): Polling, Push and Pull

- A Node System works with an internal Pulse. With this timing all data from all Nodes is calculated.
- In other words: The Pulse of a Node System is its heart.
- There are different options and settings to optimize a Node System based on this Pulses
- There are options to throttle and slow down a chain of Nodes to safe performance

Learn more about all basics, settings and options into this chapter: [Pulse\(s\): Polling, Push and Pull](#)

## Editors

### Node System Editor

The Node System Editor is your main working area where you build up and connect a Node System

Get all information about the User Interface and the Workflow here: [Node System Editor](#)

### Node System Monitor

The Node System Monitor helps you with debugging a Node System.

Select Nodes to monitor their data or observe errors.

Get all information about the Node System Monitor into this chapter: [Node System Monitor](#)

## Working With Nodes

- There are different ways how to create Nodes - all ways are straight up and deeply integrated in the software workflow of VERTEX.
- Connect different Nodes with connection lines or disconnect them
- Get to know the basic workflow, learn tricks and tips.

- Advanced users will also learn in this chapter how Node Systems behave in a Session of multiple Systems

Here we go: [Working with Nodes](#)

## Parameters and Conditions

- Most of the Nodes are working with **Parameters**. Parameters could have **different sources for their values**.

And of course, there are some advanced settings for them.

Start with learning about Parameters here: [Parameters and Conditions](#)

- Some Nodes are working with **Conditions**. With one or more of them into a Node, you are able to **program simple or even complex logical decisions**.

Get all information about the basics and also the advanced options here: [Parameters and Conditions](#)

## Composite Node

- With the **Composite Node** feature of VERTEX you can **combine an arrangement of Nodes to only one node**.
- You can then treat this **sub-composition as just one node**.
- You are able to **decompose** a Composite Node again.

Get all information about the workflow into this chapter: [Composite Nodes](#)

### 6.9.1 Pulse(s):Polling, Push, Pull

- **Pulses are the heart of a Node System**: A Pulse defines the timing with which the data of all Nodes is processed
- There is a basic Pulse, called a **Poll** (respectively Polling)
- Some Nodes are able to **Push** their data into a chain of Nodes or to **Pull** data from a chain of Nodes
- **Throttling** gives you the option to slow down specific parts of your Node System and save performance

## Pulse

- A Node System **has to collect and sync several data flows between Nodes**.
- **Data for each Node has to be calculated** and, above all, it has to be **ensured that the correct data is available at each Node** into a chain at all times.
- A Node System works with a **basic Pulse - called "Polling"** - that defines the timing for the whole Node System. On every Poll, the data of all Nodes into a Node System is calculated.
- **Some Nodes are able to generate their own Pulse**. Some can **push** their data into a chain of Nodes behind. Some Nodes are able to **pull** data from a chain of Nodes ahead.



### Too much theory?

We deliver VERTEX with default values for the 3 types of Pulses. These are basic values for timing and there are default settings for push and pull with which you can simply start working. **If you don't want and don't need to, you don't have to deal with complicated settings.**

However, with larger projects and a lot of data, the pulse setting options allow you to adjust the system load and also the signal chains very precisely.

## Advanced Options

- **There are settings for Push and Pull** with which you are able to build different **chains of Nodes in a Node System that work with their own Pulse**. For example in some cases you need more precise data, in other cases a slower Pulse is sufficient and saves performance.
- For advanced users, there are **adjusting screws for the behaviour of every Node**. This is how you can keep a good performance also for the largest and most complex Node Systems .

## Poll

- Each Node System works with a **basic "Pulse"**, an **internal timing** with which data into a Node System is sampled and processed.
- This **basic Pulse** of a Node System in VERTEX is called **"Polling"**.

## Push

- Some Nodes are able to **push their data** from their output to a chain of upcoming Nodes.

- With the **"Pulse Mode" settings** you are able to define the behaviour of a Node and how this Node generates its own Pulse.
- Use **Throttling** to for example slow down the pulse and save performance on calculating all data.

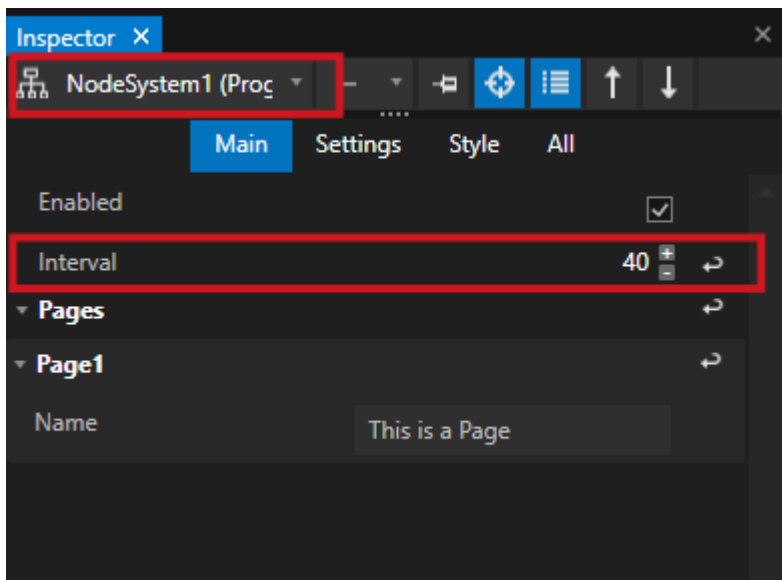
## Pull

- Some Nodes are able to **pull their data** from a chain of Nodes ahead to their Input.
- With the **"Pulse Mode" settings** you are able to define the behaviour of a Node and how this Node generates its own Push Pulse.
- Use **Throttling** to for example slow down the pulse and save performance on calculating all data.

## Polling

- Each Node System works with a **basic "Pulse"**, an **internal timing** with which data into a Node System is sampled and processed
- This **basic Pulse** of a Node System in VERTEX is called **"Polling"**

You are able to change the Polling time for each Node System in the Inspector.



**Default Value for Polling Interval**

The default value for the polling of a Node System is based on the default FPS of a sequence (25 FPS = 40 ms).

## Pulse Mode

- The Pulse Mode **determines how data is processed from a Node.**
- The Pulse Mode and the **available options depend on the Node type and the way how a Node internally works with its data**

**Passive Nodes - Filter and mathematical operations**

Mathematical Nodes that only manipulate incoming data into a chain do not have a Pulse Mode setting. They are only passive and just manipulate or filter incoming data with a defined rule set.

- Nodes with an available Pulse Mode option are **able to generate own Pulses**. In this case all data in the chain will be processed according to the pulse of this Node.

There are basically **two different Pulse Mode types** for Nodes:

**1. Push**

This kind of Nodes pushes its data into a chain chain of Nodes that is connected to its output

**2. Pull**

This kind of Nodes is actively requesting data on its Input(s).

Data is pulled from a chain of Nodes that is connected to the Input.

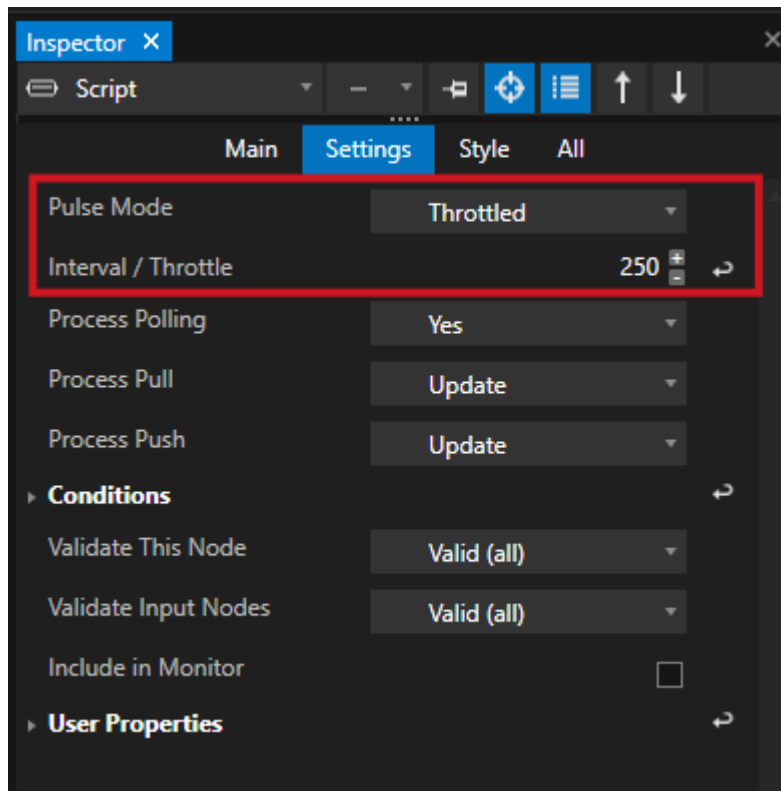
## Throttling

- For Nodes that are able to **generate their own Pulse**, you are **able to set an own timing**, an own Pulse Interval.
- **"Throttled"** means that the Node and its data is not processed more times than **specified by the defined interval**
- **Events occurring in the meantime are ignored**



### Default Values for Throttling Interval

The default value for the Throttling Interval for a Node is set to 250 ms



### Throttling saves system performance

Throttling gives you a lot of adjusting screws to optimize the performance of large Node System.

Into a Chain of Nodes you have the possibility to save performance and reduce calculations. Depending on your Project this could be on different places: less CPU usage because throttling brings less calculation, less Network traffic, less GPU usage.....

### Simplified Example:

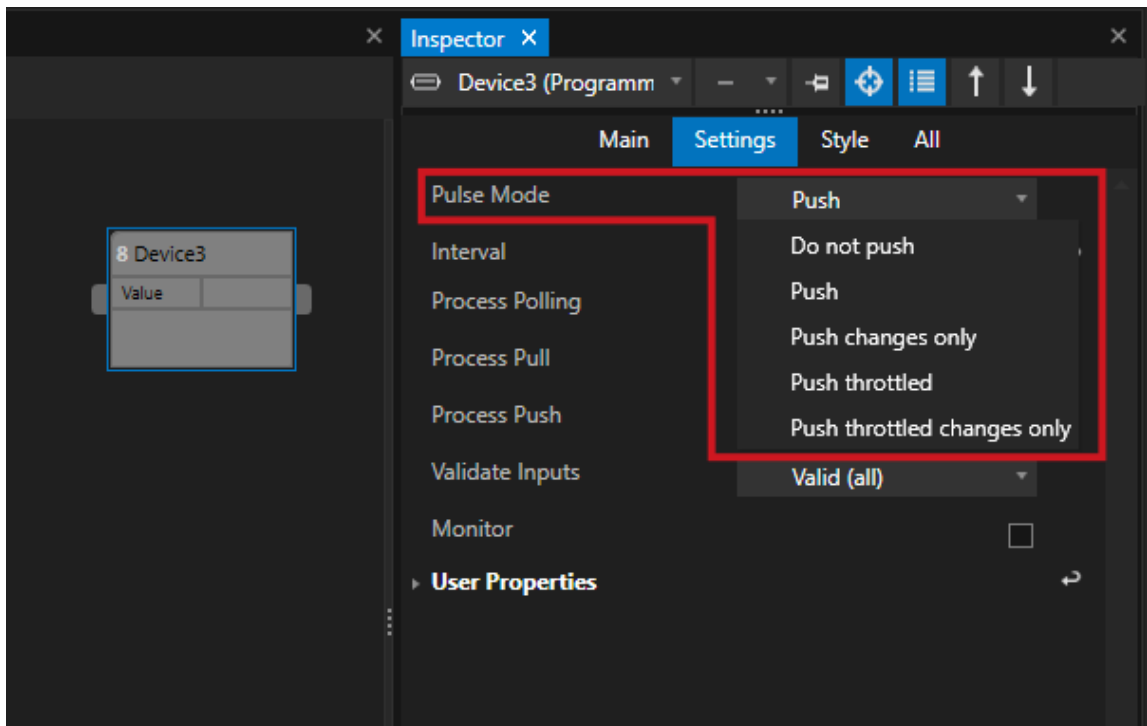
An [UDP Device](#) should receive a status from an UDP-sending 3rd-party Device. This status should be displayed as label of a [ControlView](#). Because the status only changes between "i am here" and "i am off" - it is sufficient to calculate and change it for e.g. every second. In this case it make sense to throttle the Node from the TCP receiver. This Node should push the data throttled with an interval of 1000ms.

For more detailed options, please also read the follow up chapter "[Settings](#)".

## Settings

For Push or Pull there are different settings, to fine adjust the Pulse Mode

### Push



#### **Do Not Push:**

*This Node does not trigger a Push*

#### **Push:**

*This Node triggers a Push (default setting)*

#### **Push changes only:**

*This Node triggers a Push, but only if the output values have been changed*

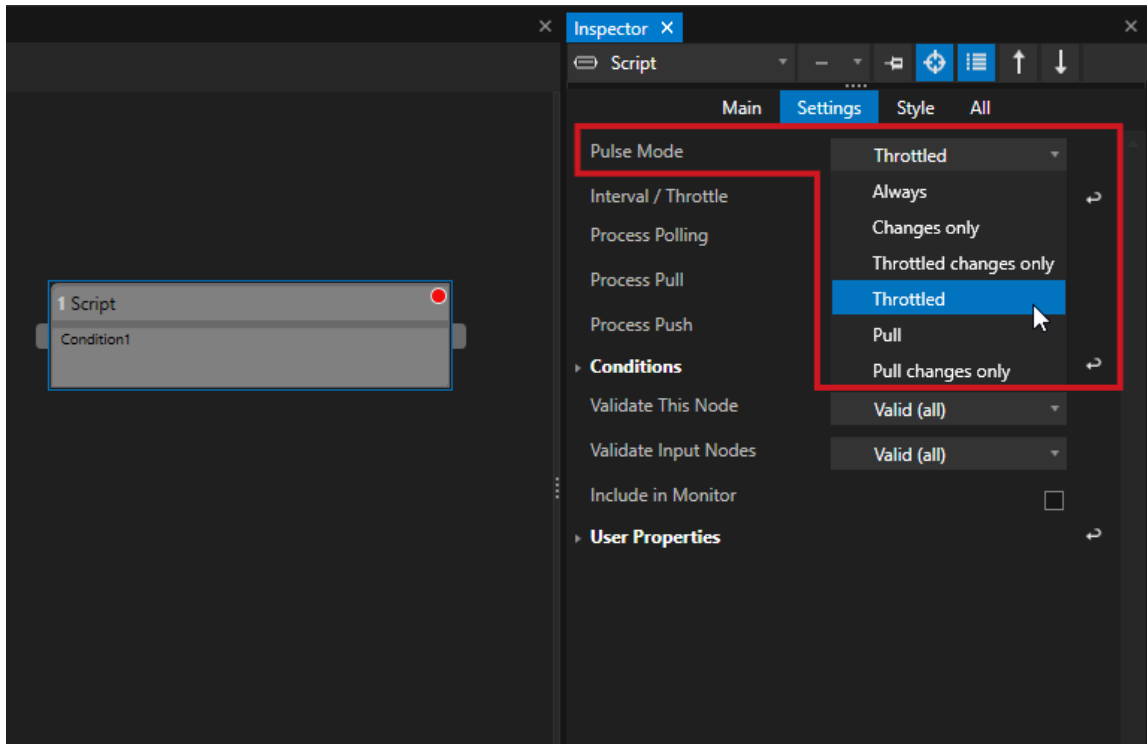
#### **Push throttled:**

*This Node triggers a Push but only with the throttled interval that is set*

#### **Push throttled changes only:**

*This Node triggers a Push with the throttled interval that is set and only when the values on its output has been changed*



Pull**Always:**

Process this Node and always execute - regardless of changes and without any throttling

**Changes only:**

Process this Node but only execute, if the the resulting output values have changed

**Throttled changes only:**

Process and execute this node with the throttling interval but only if the resulting output values have been changed

**Throttled:**

Process and execute this node with the throttling interval

**Pull:**

This Node triggers a Pull request with the Interval that was set (-> throttling interval)

**Pull Changes only:**

This Node triggers a Pull request but only if the resulting output values have changed

## Advanced Settings

- For Advanced users there are options to define a **Node's behavior for each o the three Pulse types:** Polling, Push and Pull
- You are able to set, **if and how a Node should be processed** for those three Pulse cases

**Process Polling:**

Defines the behaviour of a Node on the main polling pulse

**Process Push:**

Defines the behaviour of a Node for Pushes

**Process Pull:**

Defines the behaviour of a Node for Pulls

**Settings:**

**No:** This Node will not be processed

**Yes:** This node is processed, but its input values (if connected to outputs of other nodes) are not updated.

**Update:** This node will be processed - but it specifies that its inputs will be updated first, by processing upstream nodes as well.

## 6.9.2 Node System Editor

- The Node System Editor helps you to **create, arrange, control and order** your Nodes
- With different **monitoring tools** you are able to get an overview about the data flow or the state
- **Snapping tools, rulers** but also **layout controls** like labels or tile boxes help you to keep the overview into your Node System

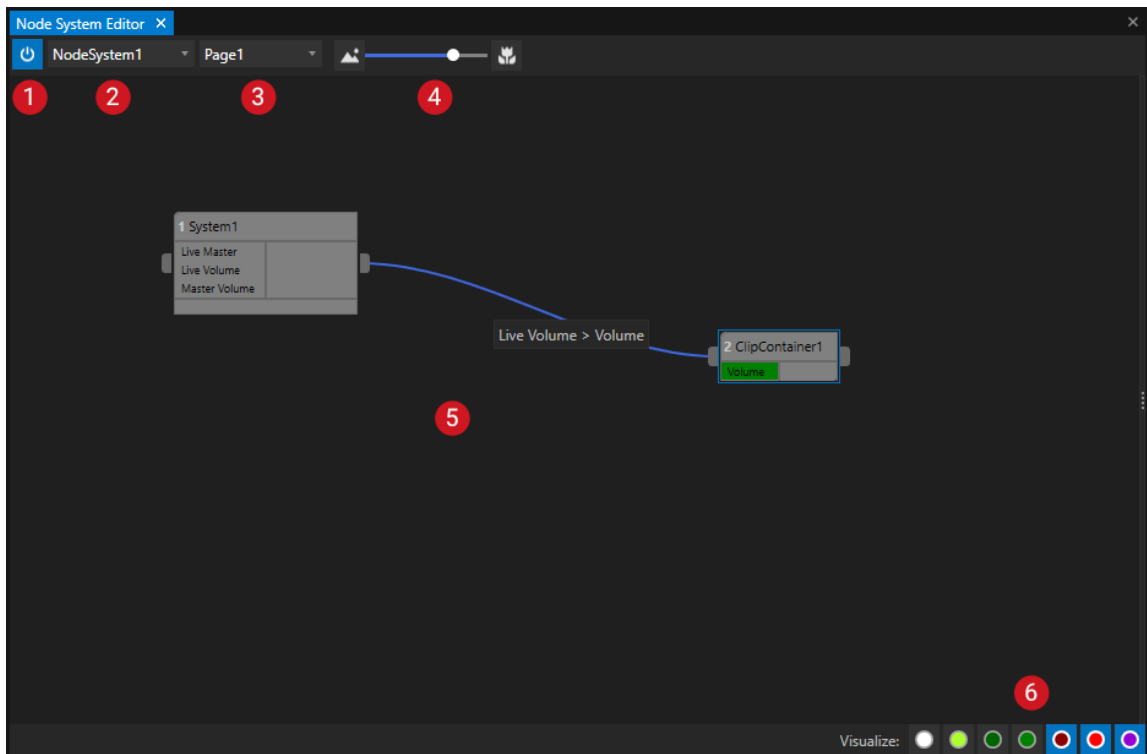
## User Interface

**Create an new Node System first:**

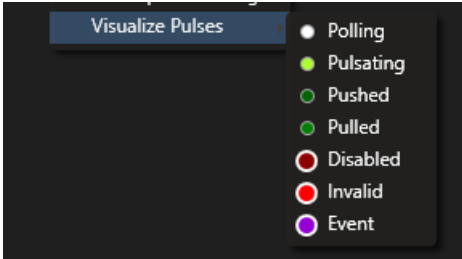
- Go to the Main Menu on top, select "Create" there and "Node System"  
or
- Go to the "Node System" category into Project Explorer, open the context menu by a right-click and create a new Node System

**Open a Node System Editor:**

- Go to Main Menu > Windows
- Open a new Node System Editor



1	<b>Enable/Disable Node System</b>	Default: Enabled Click to disable the selected Node System - also for debugging or if system load increases too much
2	<b>Select a Node System</b>	Select a Node System or a <a href="#">Composite Node</a> to show this into Editor
3	<b>Select a Page</b>	Select a Page of your Node System Default: only one Page is available - more are displayed if they have been previously created via e.g. the context menu in the Node Editor
4	<b>Zoom in/out</b>	Zooms working area in and out Alternative: use Keys "+" and "-" on your Keyboard
5	<b>Working Area</b>	Main working area to create and wire your Nodes
6	<b>Visualize Pulses</b>	Different options to monitor the data flow and the data pulses of your Node System: Blinking Dots are displayed into every Node. They visualize if/when data is pulled, pushed, if nodes are disabled

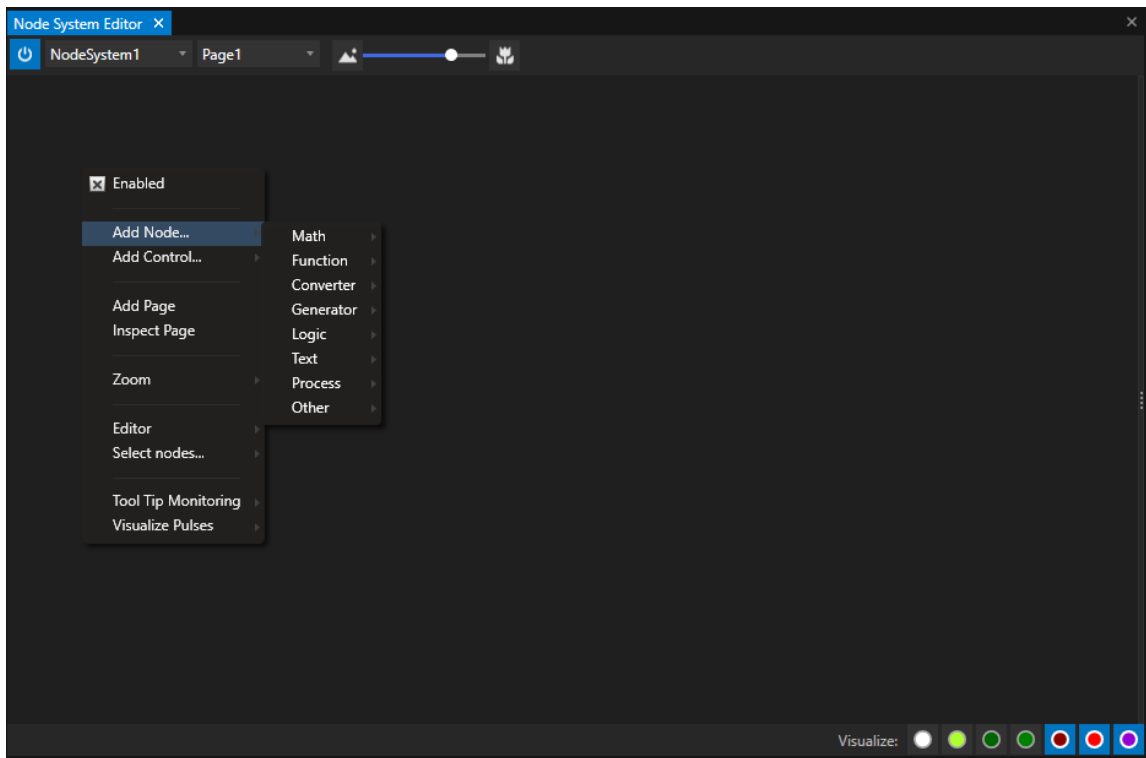
		<p>or actions are triggered.</p> <p>The Pulse visualization help you to debug your Node System.</p> <p>You are able to select and enable/disable the selected types by color.</p> <p>The tool tips help you to explain the different types.</p> <p>Also could be switched on/off into Context Menu</p>  <p>The screenshot shows a dark-themed menu titled 'Visualize Pulses'. It contains seven items, each with a colored circular icon to its left: 'Polling' (white), 'Pulsating' (yellow-green), 'Pushed' (green), 'Pulled' (light green), 'Disabled' (red), 'Invalid' (red), and 'Event' (purple).</p>
--	--	---

## Context Menu

With a right-click into an empty space of the working area, you open the context menu.

This menu gives you access to the most common actions like:

- Add a Node
- Add Controls as visual helpers for your node layout or labels
- Add and Inspect Pages
- Do zoom settings
- Select one of your nodes from a list
- Do settings for the Node System Editor, the editors behaviour or for monitoring data



**i** **Context Menu for Nodes**  
 If you right-click on a Node, an extended context menu opens.  
 This contains additional options for the Node, but also the settings for the Node System Editor.

## Create and Connect Nodes

Node Systems are deeply integrated into VERTEX.

The **workflow** of the Node Systems Editor corresponds to **the one you already know** in other places in the software:

- You e.g. can create Nodes by drag and drop items from Project Explorer or from the Playback Editor to the Node System Editor
- The same works for Properties from the Inspector (you might already know this from e.g. the [Wiring](#) or [Triggering Editor](#)). Just drag a Property from Inspector to the Node System Editor and a context menu opens to create a Node

There are a lot of other **possibilities and options**. All you will find summarized in the extra topic [Working with Nodes](#)



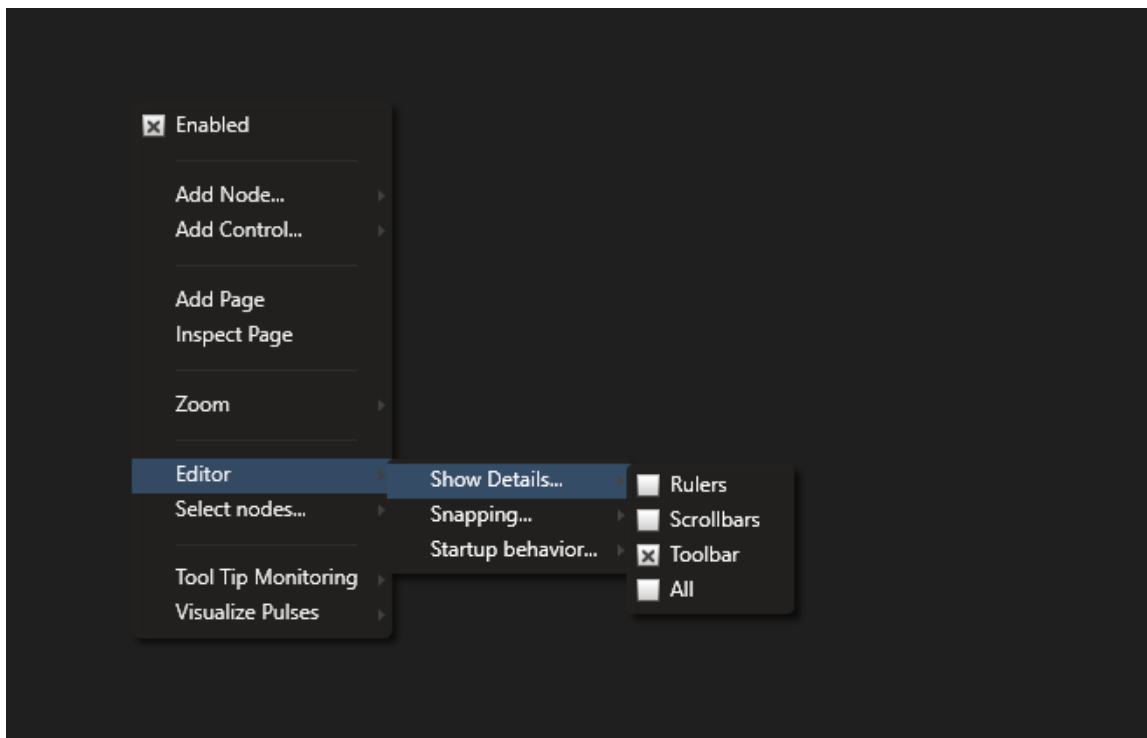
#### Expand/collapse Node(s) with a double click

Double-click with your mouse on a Node to collapse or expand it. Select multiple Nodes, hold the SHIFT-Key and double-click to expand or collapse multiple Nodes.

## Editor Settings

There are different options for the Node System Editors **layout**, the **snapping** and **startup behaviour**.

- Right-click into the working area
- open the Context Menu
- select one of the entries and
- set your enabled or disabled flag there.

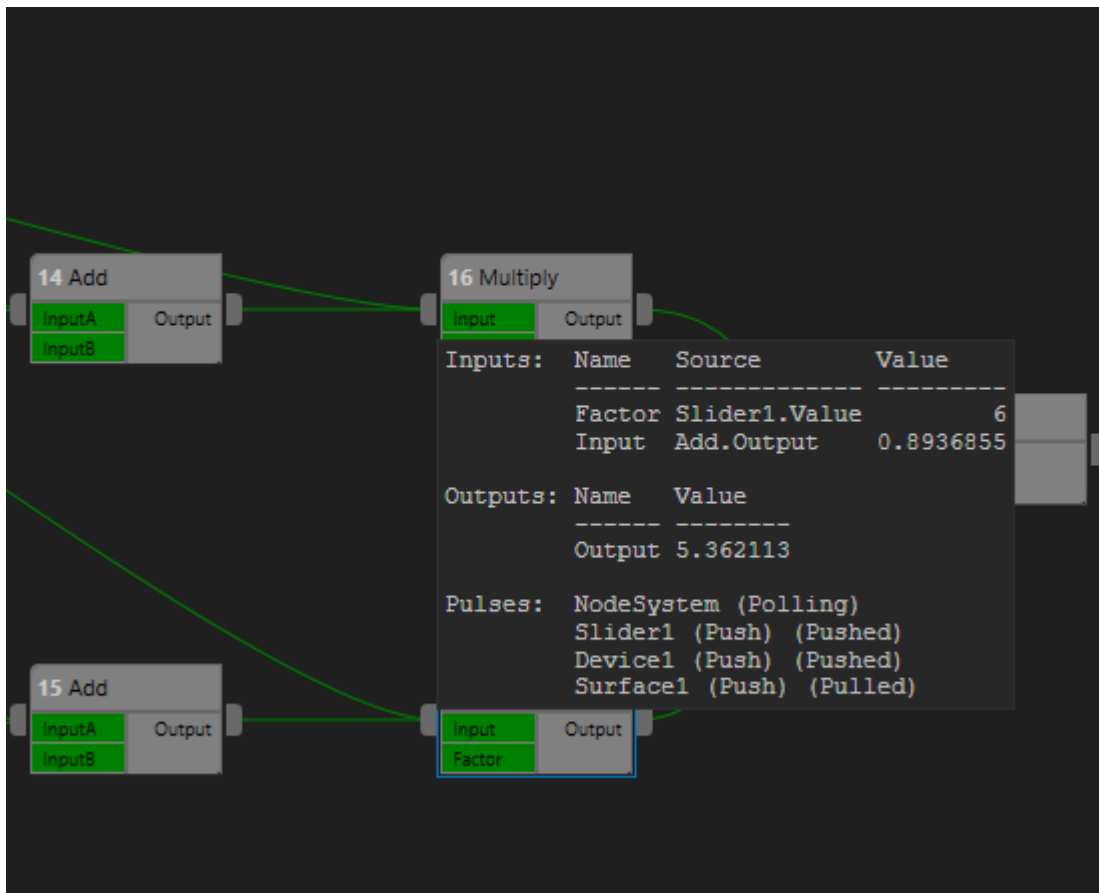


## Data Monitoring

The Node System Editor offers you a **Tooltip Monitoring** to quickly check:

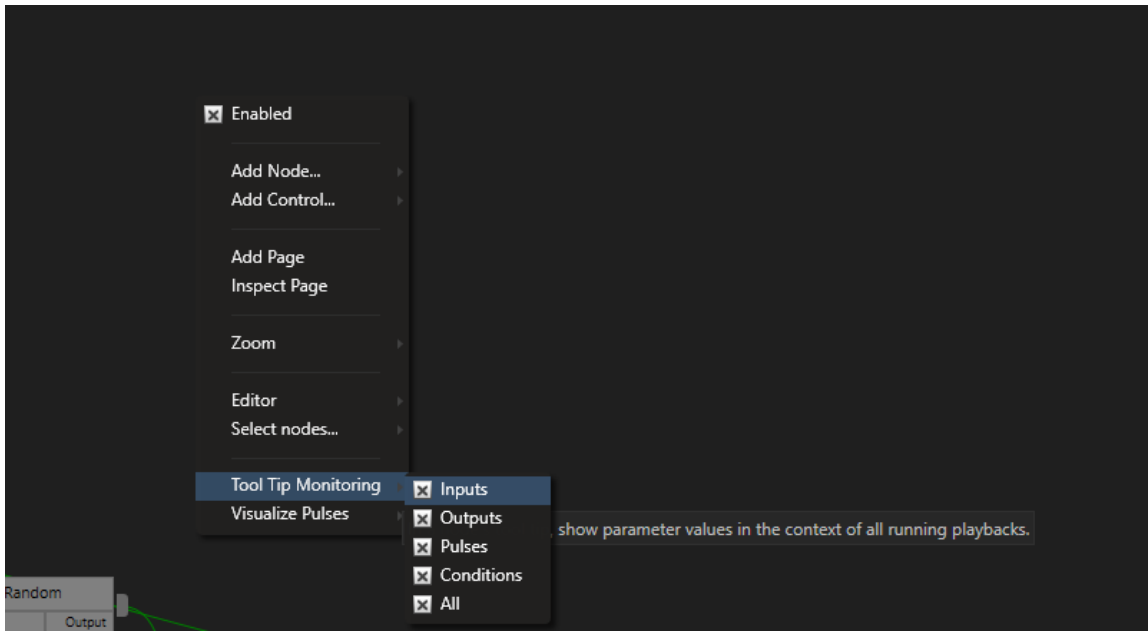
- Input and Output Data of a Node
- Conditions of a Node
- Pulse types of a Node

Just **move your mouse above a Node**. The Tooltip Monitoring **pops up** and **shows** you the **current values** and **all information**.

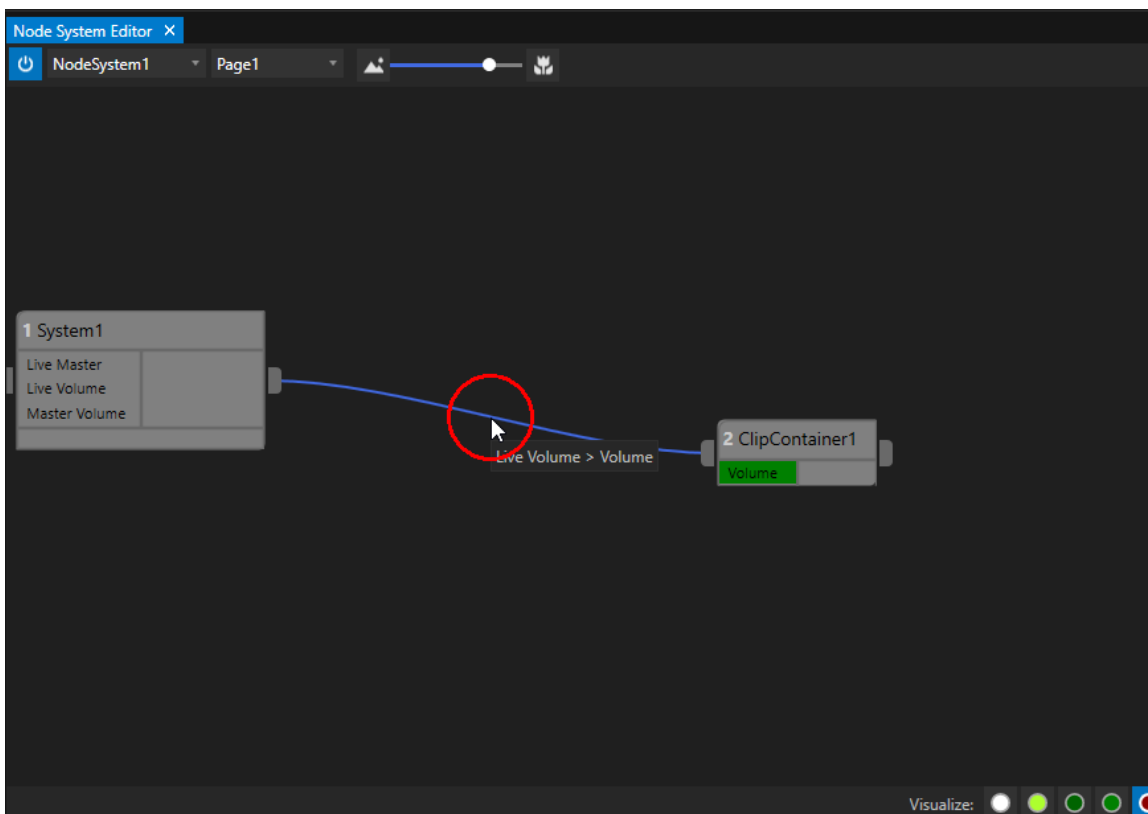


To specify and customize the information that should be displayed into the tooltips:

There is a entry into your Context Menu where you are able to select the information that should be displayed.



When you move the mouse over the connection lines the connected properties will be displayed as a tooltip:





## Pulse Monitoring

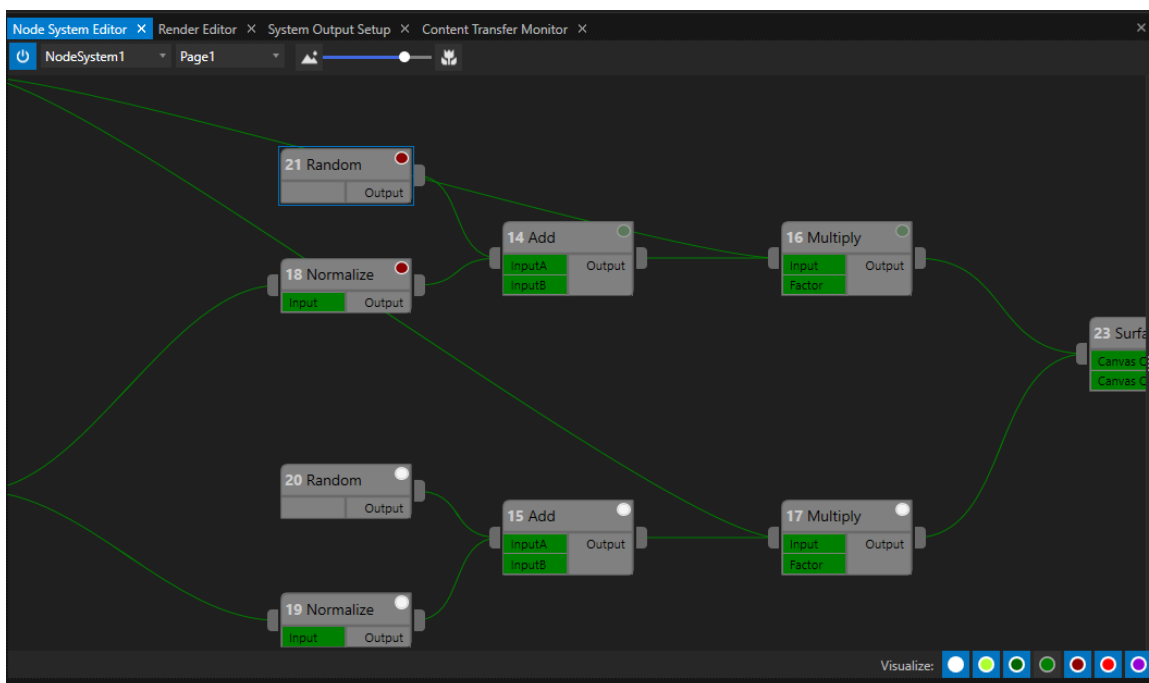
For testing, debugging or just to get an overview our Node System Editor offers you different types or data pulse visualizations for your Nodes.

Data is pushed to Nodes and/or pulled from Nodes with a Pulse. Sometimes it could be helpful to animate this different types of Pulses or the current Node state.

The build-in Pulse visualization **helps you to e.g. check**

- if/when data is **polled**
- if or when data is being **pushed**
- if or when data is being **pulled**
- if data is **invalid**
- if the node is **disabled**
- or when an **event** - like a script execution **is triggered**

The pulses are displayed as blinking dots into every Node.



There are two ways to set :

- Visualize menu at the footer/right-bottom of the Node System Editor
- "Visualize Pulses" into the Context Menu (open with a right-click into the Node System Editor)

Both ways always enable the pulse visualization type **globally for all Nodes**.

### 6.9.3 Node System Monitor

- The Node System Monitor helps you with **debugging your Node System**
- Monitor and observe **incoming and outgoing data of a selection of Nodes**
- Get informed about **wrong incoming data and/or errors**

## User Interface

There are **2 ways to open** a Node System Monitor

#### 1. Open as an extra editor window

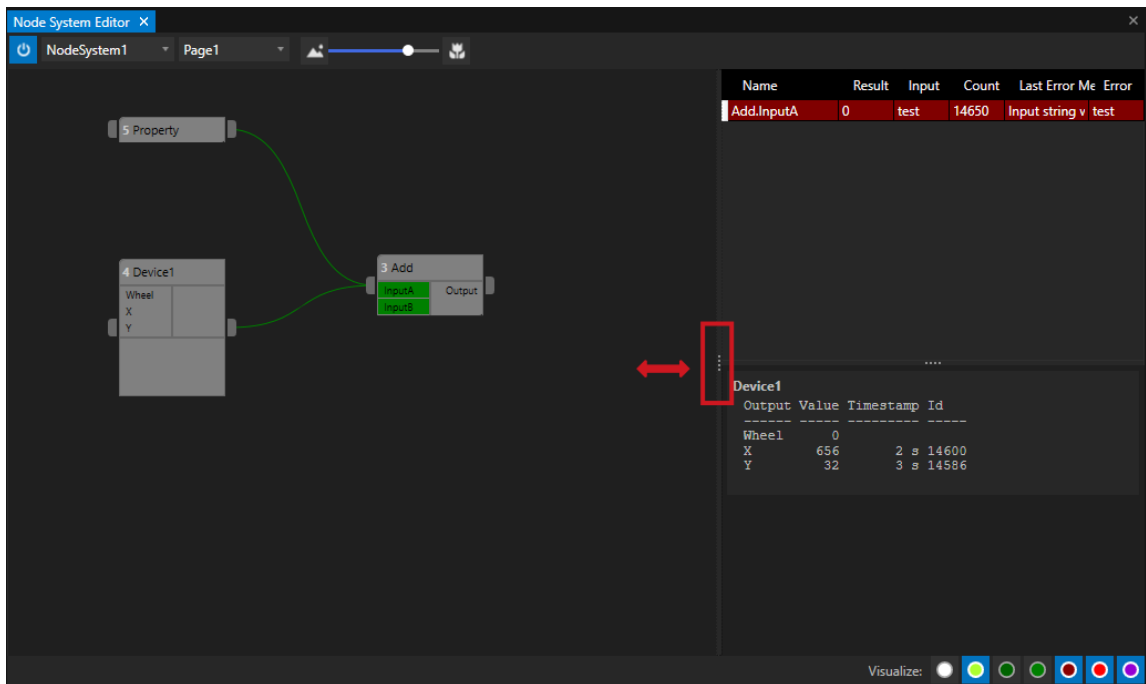
As for all other editors and windows:

- Go to the "Windows" tab at Main Menu on top
- Open a new "Node System Monitor" window

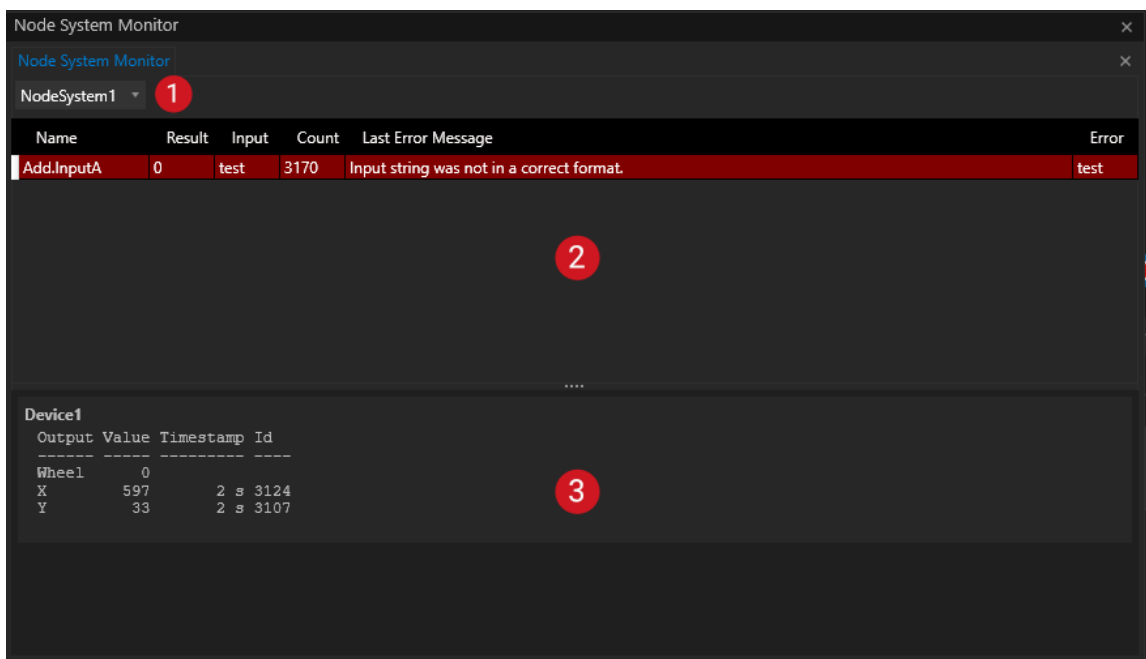
#### 2. Quick Access - move out of Node System Editor

If you work with the Node Systems Editor and want to have just a quick access to the Node System Monitor:

- Move your mouse to the 4 dots on the right side of the Node System Editor window
- Drag to the left with pressed mouse button
- The Node System Monitor window opens as a "split screen"



## Window Layout

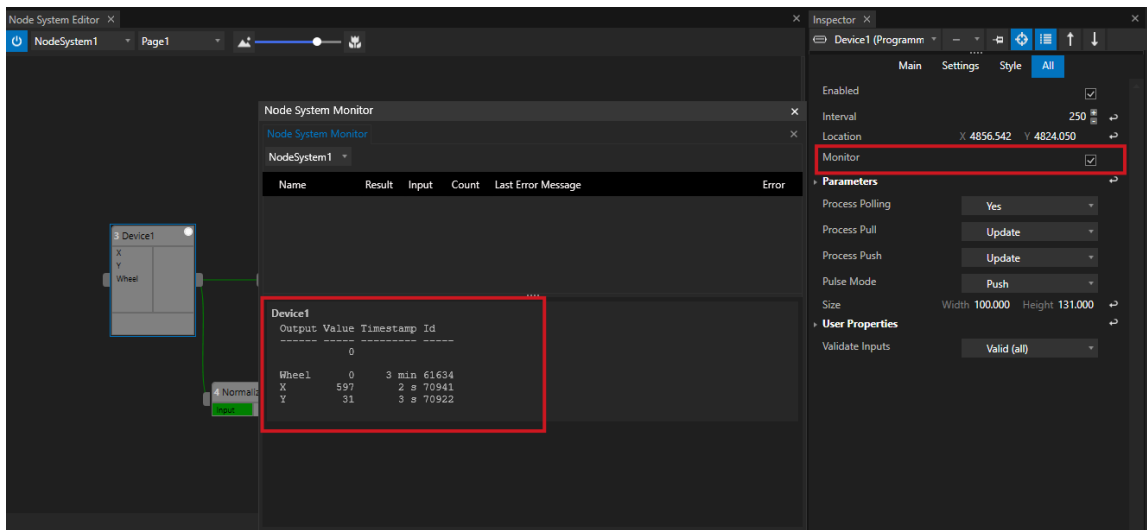


1	<b>Select Node System</b>	Select one of your Node Systems which you want to monitor
2	<b>Error Log</b>	<p>Displays all data errors that occur during your Node System is running.  <b>Colored in Red</b> if an data processing error occurs.  <b>Colored in green</b> when correct values arrive again after a previous data error.</p> <p><b>Columns</b>  <b>Name:</b> the name of the Node or <u>Parameter</u> where the error occurred  <b>Result:</b> the value that this <u>Parameter</u> currently outputs  <b>Input:</b> the currently incoming data  <b>Count:</b> how often the current input has already been used - number of internal calculating pulses - low number: error has just happend, high number: incorrect value has been on for a long time  <b>Last Error Message:</b> last error message  <b>Error:</b> input that triggered the error.</p>
3	<b>Node Monitoring</b>	<p>Displays the outputs of preselected Nodes.  All Nodes where the <u>monitor flag is set in the Inspector</u> are displayed here.</p> <p><b>Columns</b>  <b>Output:</b> Output Parameter name  <b>Value:</b> Current value of the output Parameter  <b>Timestamp:</b> Time that has passed since the Parameter was last changed  <b>ID:</b> Pulse id where the Parameter's value was last changed</p>

## Add a Node to the Node System Monitor

- Select a Node into the Inspector
- Search for the "Monitor" Property
- Enable the Monitor flag in the Inspector

All Nodes with enabled "Monitor" flag are listed with its output information into the Node System Monitor

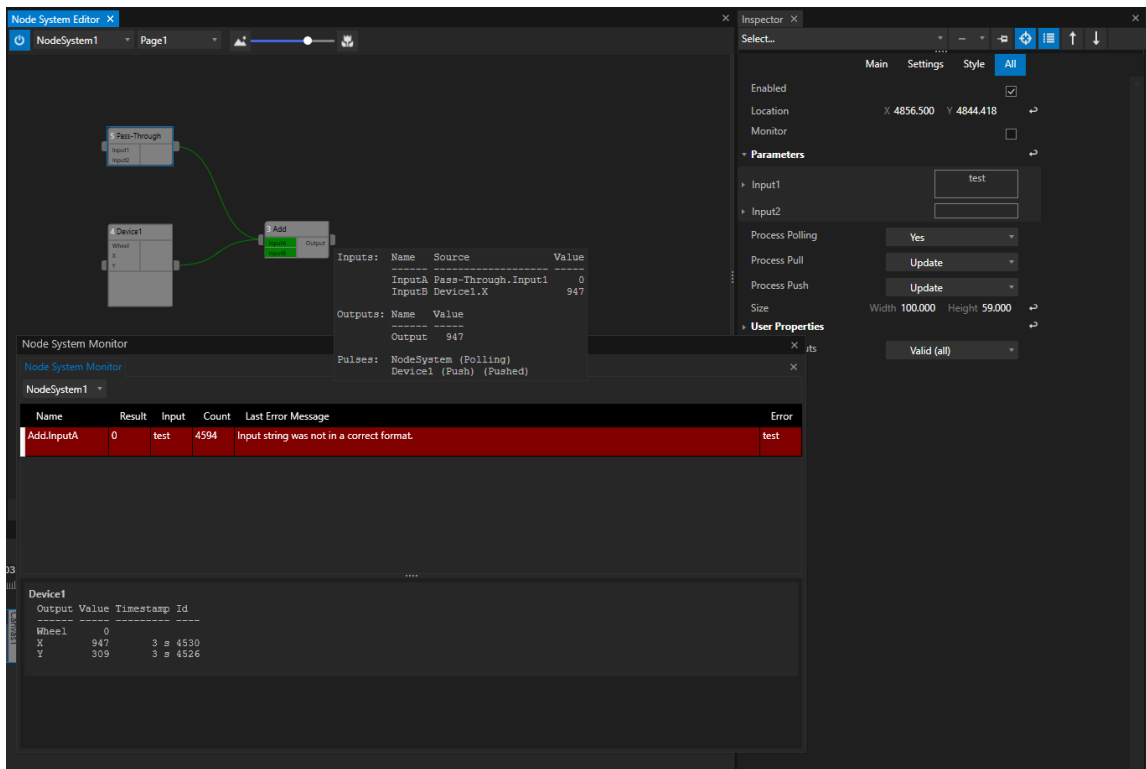


## Interpret Errors

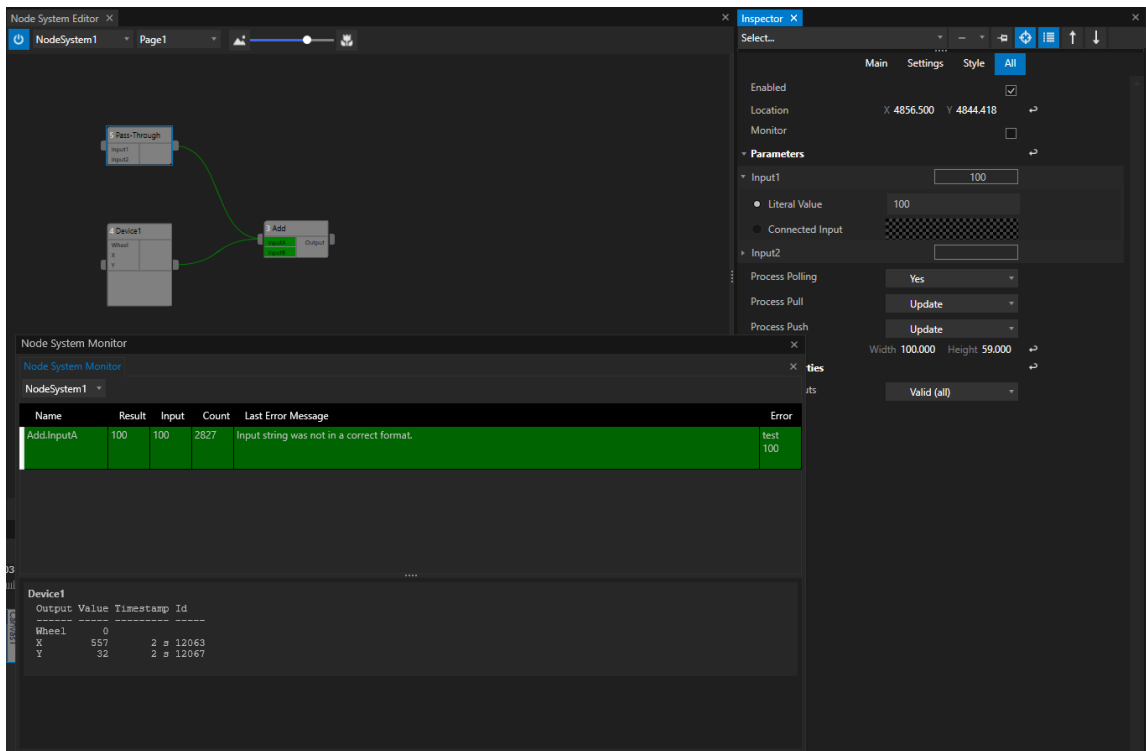
The Error log of the Node System Monitor should help you to interpret and collect errors that are thrown while your Node System is "running" and calculates data.

**There are 3 types of entries possible:**

1. **No entry:** There is currently no error and no errors have occurred since the Node System was started.
2. **A red entry:** There is currently an error. There are incorrect values at a Node input.
3. **A green entry:** An error has occurred in the past and an incorrect value has been applied to an input. This value is now correct again. The error is nevertheless recorded.



The example of the screenshot shows a Node for a Mouse Device that is connected to a Pass Through Node with two Parameters added: The values of the Parameters are passed through from input to output (In this example case the pass through device is only used to generate some values and connect them to the input of the "Add Node") The "Add Node" expects two decimal numbers as an input. The Mouse Device Node gives an decimal x coordinate as an Input B. Input A of the Add-Node is connected to an Output of the Pass-Through Node. The value of its output is a text - not a decimal number. An error is thrown.: The Input is not in the expected range.



Pass-Through Node: The value from the Input 1 Parameter ) was changed from a text "test" to a decimal number "100". Now the expected value range applies on the Input of the "Add" Node. The calculation of the "Add" Node can be processed.

In the Node System Monitor the line with the logged Error changes to a green color. The old wrong text value "test" still is listed above the correct new value "100"

?

**Errors are "expensive" for real-time node systems**

Node systems that calculate and process data directly ("real-time") are built to calculate large amounts of data - over and over again. Errors are usually very "expensive" - they block resources that are needed to calculate other data. They should not be undetected and therefore they even remain as "green colored" entries in the Node System Monitor until the Node System is stopped and restarted. Sometimes it is helpful to observe such cases, especially when the error has "become obsolete" because valid values have arrived in the meantime.

### 6.9.4 Working with Nodes

- Learn the **different ways** how to create Nodes in VERTEX
- There are various options how to **combine Parameters from different Properties in a single Node**
- **Connect** Nodes and enable a **data flow** between them, **disconnect** them or just **deactivate** the data connection

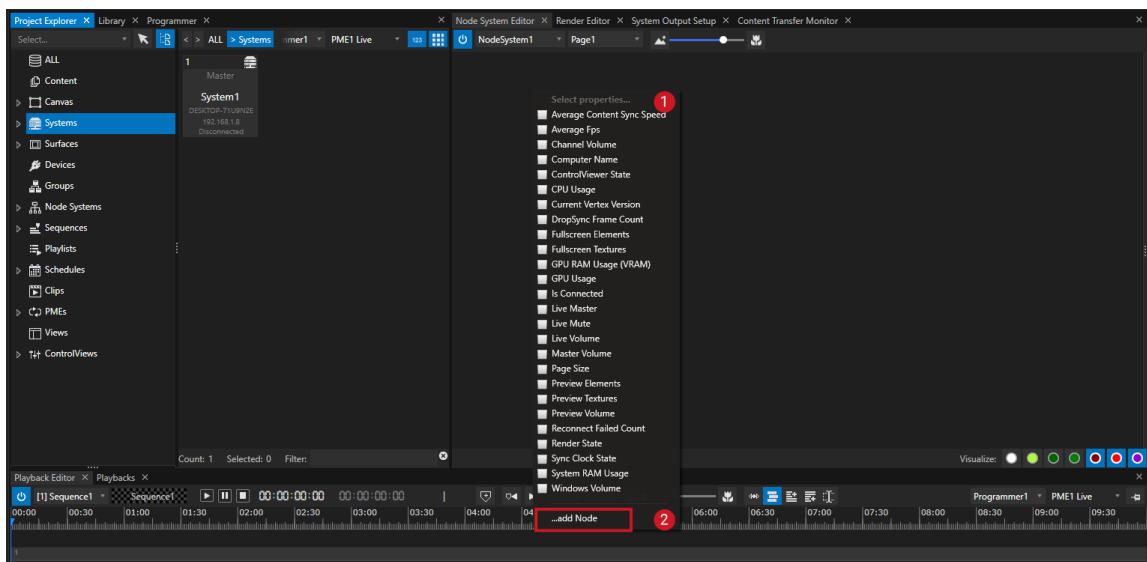
## Create a Node

VERTEX offers you **different ways to create a new Node** into a Node System. All ways are **deeply integrated into the VERTEX UI** and adapted to the workflow of our software

### Drag an item from Project Explorer to the Node System Editor

Create a new Node by simply drag and drop an item from Project Explorer to a Node System Editor:

- **Select an item** (e.g. Content, a System, a Surface) in the Project Explorer
- **Drag this item** with your mouse to **the Node System Editor**
- A **context menu opens** where you can **choose from a list of all available properties**,
- **Select one or multiple properties** from the list (1)
- **Confirm your property selection with "Add Node"** (2)
- A new Node with the selected properties is created

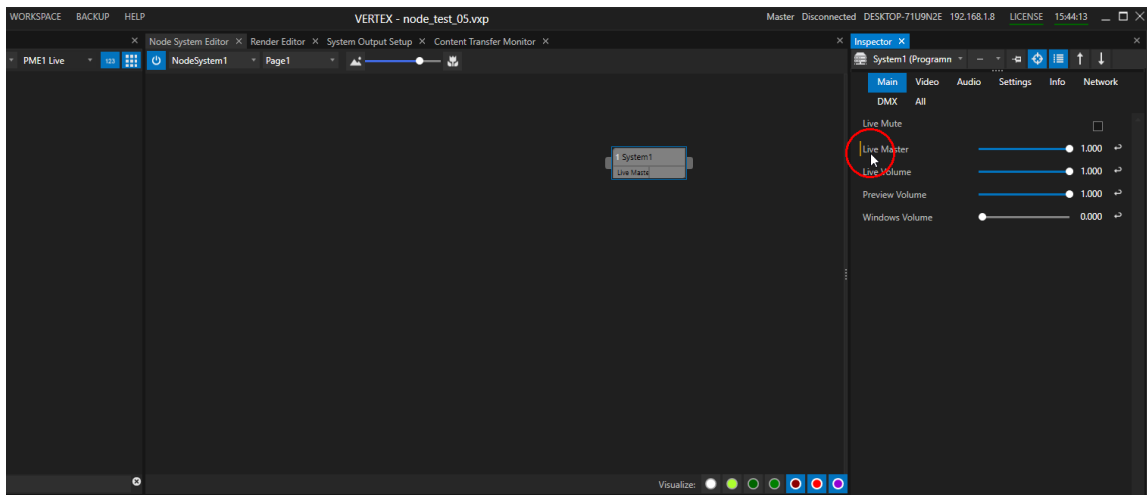


### Drag Properties from Inspector to the Node System Editor

- *Select an item*
- *Go to the Inspector*



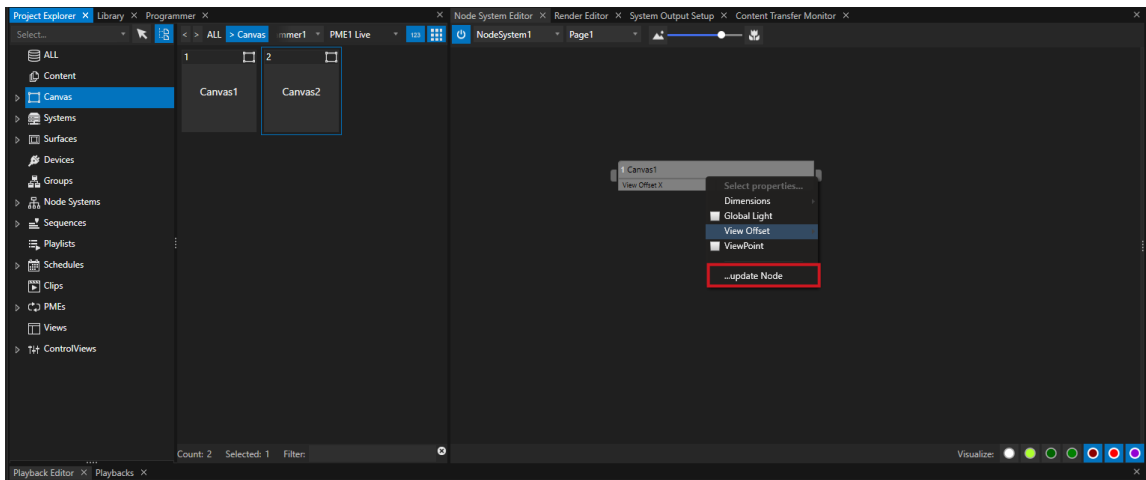
- **Select a property** in the Inspector.  
Like you may know from [Wiring](#) or [Triggering](#), properties that could be used are marked with a **yellow vertical line**.
- **Drag this property** with your Mouse to the Node System Editor
- A **new Node** with the selected property as parameter is created



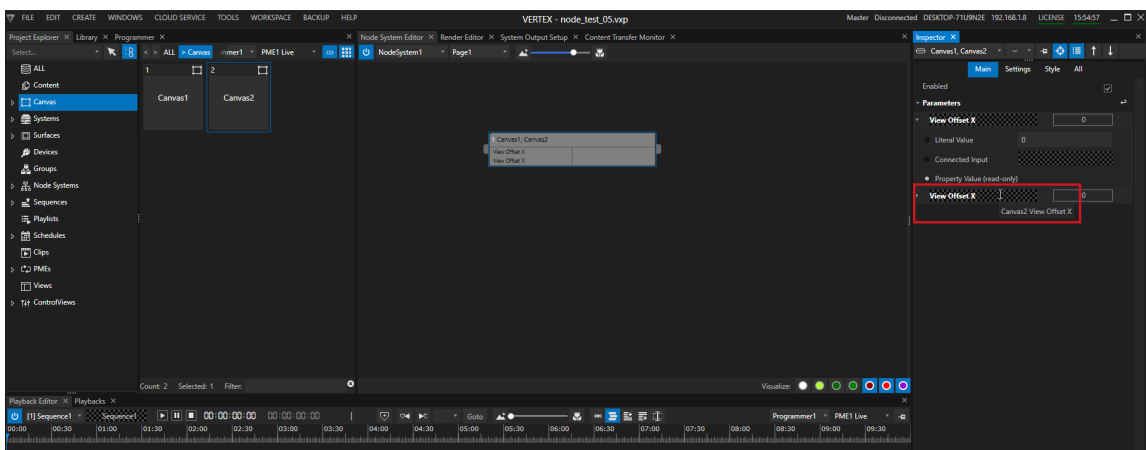
## Combine different properties and create custom Nodes

VERTEX gives you the option to create **custom Nodes** that **combine Properties** from different items.

- **Create a Node** with properties **from a first item** (e.g. Canvas 1)
- Select a **second item** (e.g. Canvas 2)
- Drag a **property of this item from Inspector to the already created Node**
- Or drag the item from Project Explorer to the already created Node. When the mouse button is released, a context menu pops up where you can select the properties you want to add.
- Select **"update Node"**
- The already existing Node is updated and converted to a Node that combines properties from different items (e.g. Canvas 1 and Canvas 2)



**i** **Show full property source of a Node Parameter**  
 For a better and cleaner overview - the property fields of a Node in the Inspector show reduced information: They only shows the property name but not the full origin. When working with combined Nodes, a **tooltip displays you the whole property source**. To show this tooltip, **hover with your mouse over the parameter field into Inspector** or over the parameter field of the Node into Node Systems Editor.

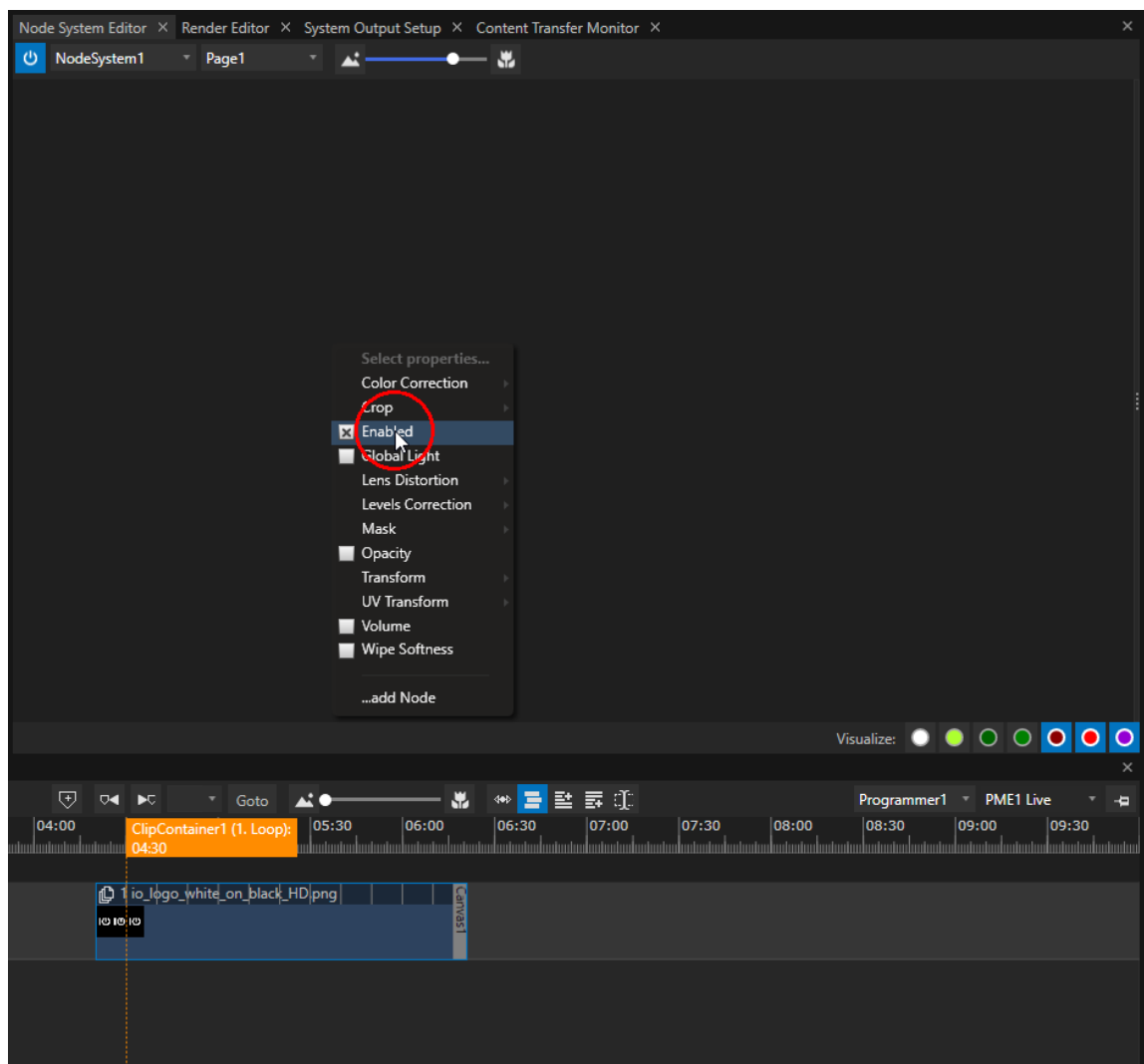


Tooltip with full source of a Parameter property

## Drag a Clip Container to the Node System Editor

- Select a Clip Container into Playback Editor

- Hold **CTRL + SHIFT** on your keyboard
- **Drag the Clip Container** with your mouse **from the Playback Editor to the Node Systems Editor** (continue to hold down CTRL+SHIFT)
- **Release the mouse button and the keys** and a **property selection menu is displayed**
- Select the properties you want to have as parameters for your Node
- Confirm with **"Add Node"** on the bottom of the context menu window



## Drag and Drop from any other Editor

Creating Nodes via drag and drop is also possible for other some other VERTEX Editors.

Hold **CRTL** and **SHIFT** and drag e.g. a Control from the Control View Editor to the Node System Editor.

Or Drag an DMX-Routing from the DMX-Routing Editor to the Node Systems Editor



### Devices and Nodes

Remember that you first have to add [Devices](#) from the [Library](#) to your Project.

Once added, you are able to drag them from the Project Explorer to the Node System Editor.

Creating Nodes directly out of the Library will not work.

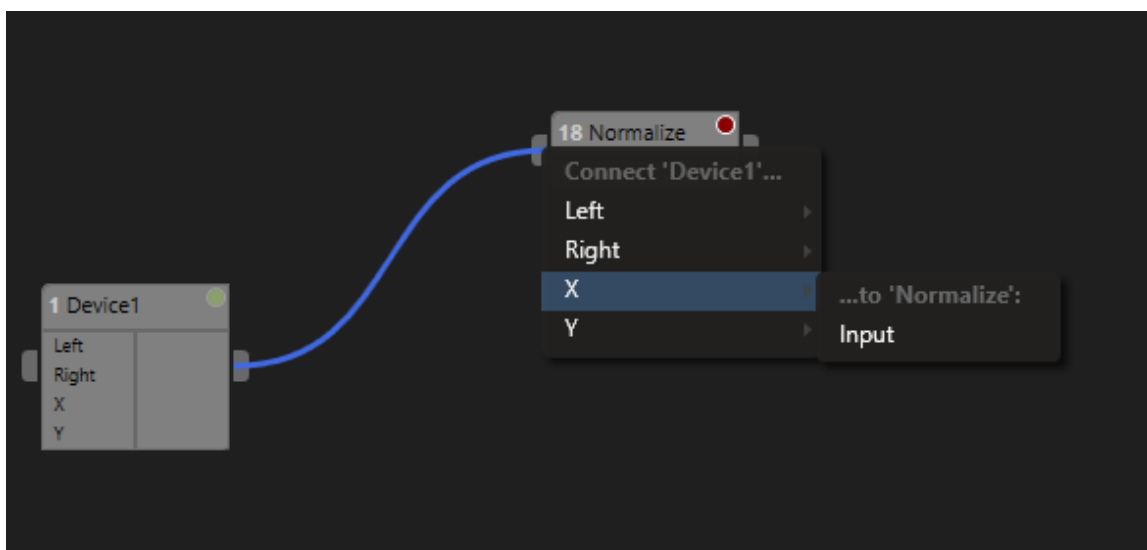
## Connect and Disconnect Nodes

To enable a data exchange between your nodes, they have to be connected with a Connection Line.

Without Connection Line there is no data flow between an output of a Node and the input of another Node.

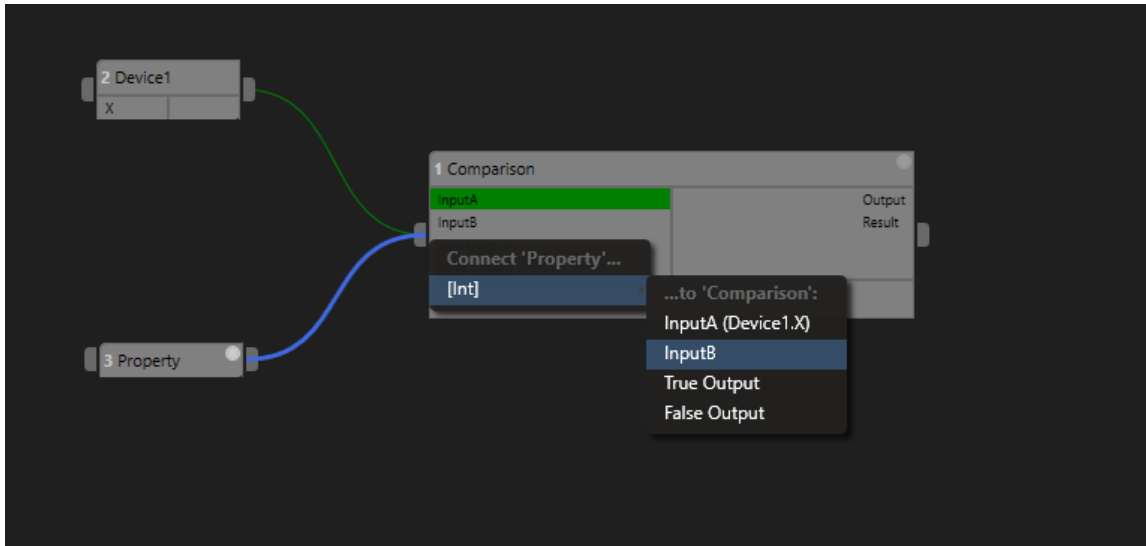
Connections are elementary for a Node System to work.

### Connect



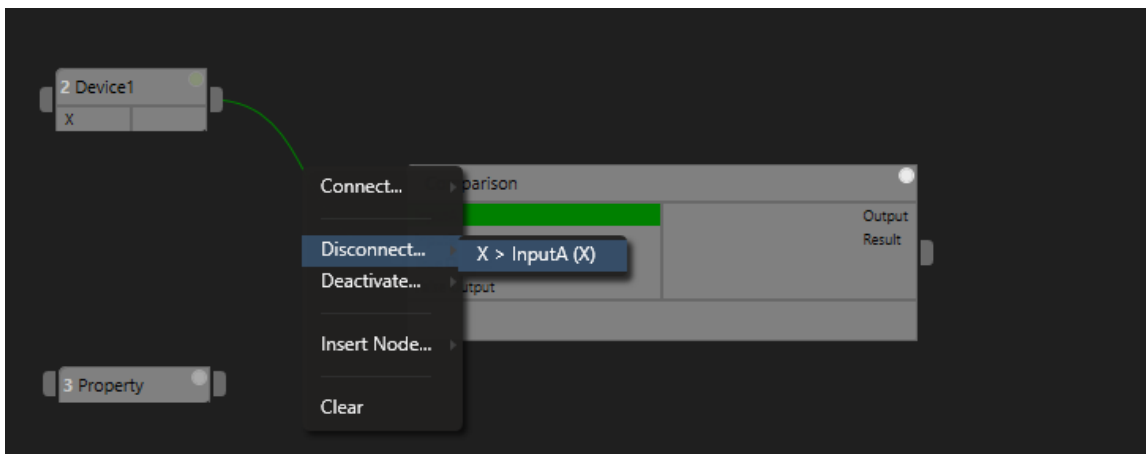
- Select the **Output pin** on the right side of a Node and **move your mouse**
- A **blue Connection Line** is drawn
- **Drag the mouse** with the blue Connection Line to an **Input pin** of another Node (on the left side of a Node)
- When you have reached the Input pin, a **context menu** opens:
  - Select the **Output from the first Node** into this context menu and
  - then select the **input or condition** of the target Node

- When the connection is successful, the connection line turns into green color and the Input Parameter of the second Node is highlighted green



## Disconnect

- **Right-Click** with your mouse on a connection line
- The **context menu** opens
- Select **"Disconnect..."**
- Select the **Connection** you want to disconnect





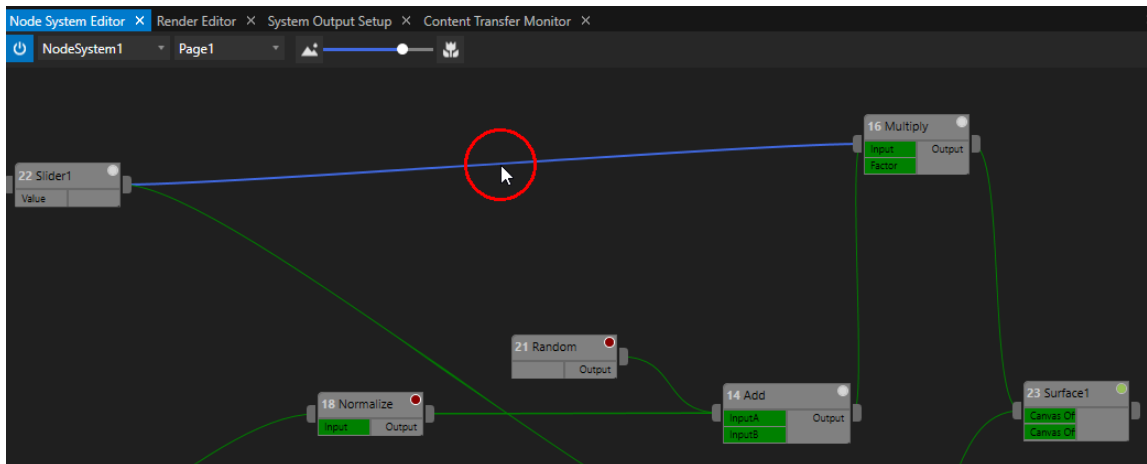
### **A single Connection Line and more than one data connection**

No matter how many Parameters are wired between two nodes, there is always only one Connection Line. If there is more than one parameter connected, the disconnect menu offers you several options. If there are e.g. two data connections between Nodes and only one of this connections is disconnected, the green Connection Line is of course kept

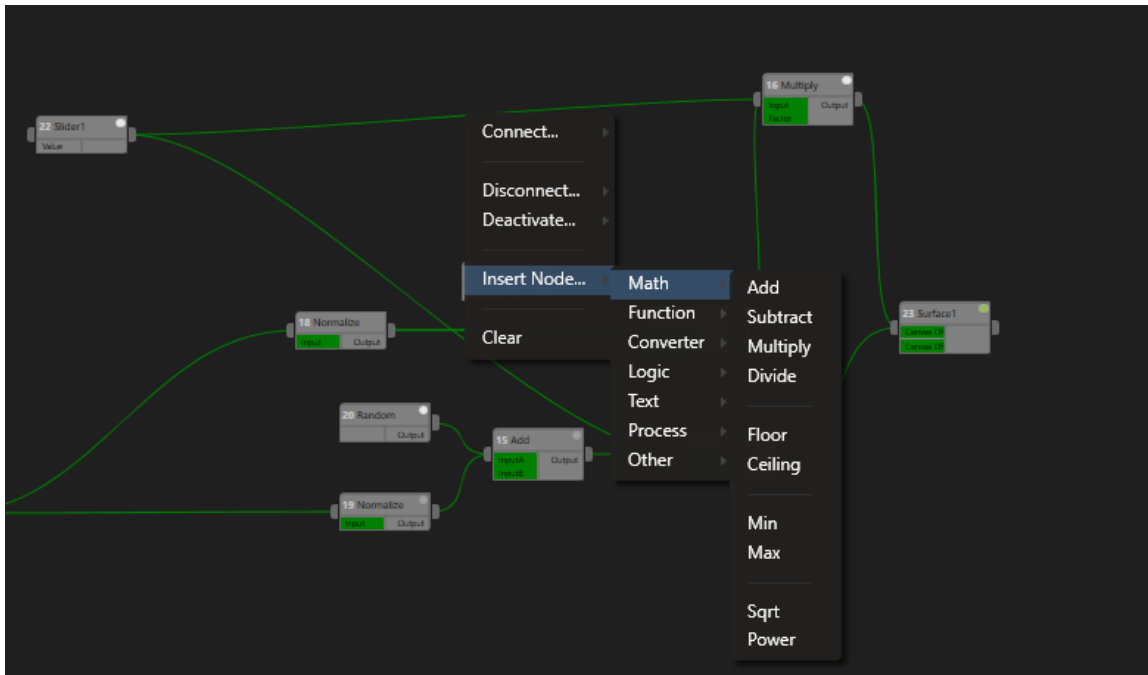
## Insert a Node

- It is possible to insert a Node between two already connected Nodes.

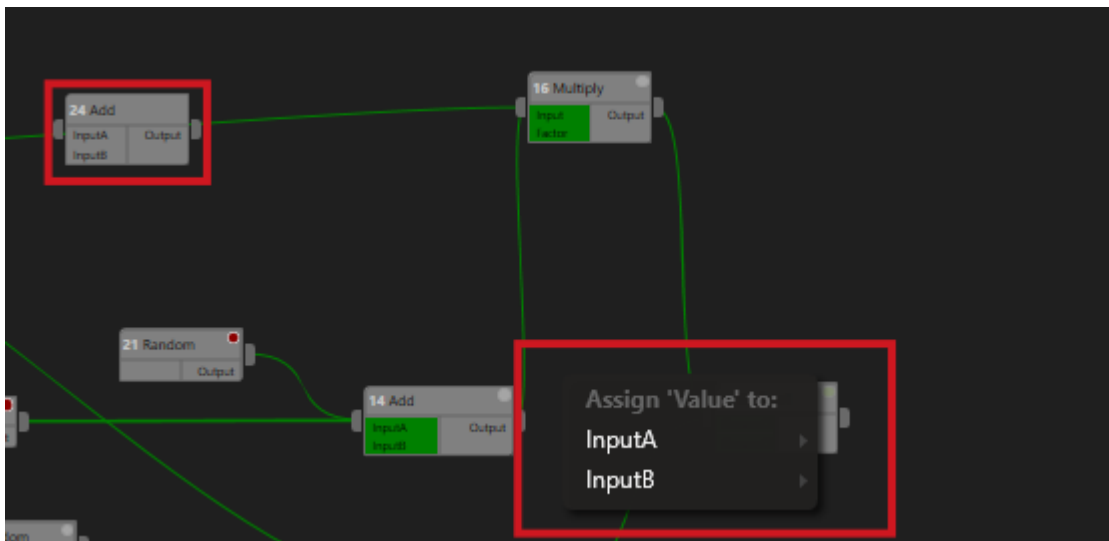
### Steps



1. **Right-Click** with your mouse on a connection line between two nodes
2. A **context menu** opens
3. Select **"Insert Node"**
4. Select a Node you want to insert



- The selected Node will be inserted between the two initial Nodes
- Depending on the inserted Node type, **another context menu opens where you can assign the values to the input parameters** of the new and inserted Node



Depending on inserted Node type: Last step is to connect the data to the inputs of the inserted node. After a new node is inserted between, a context menu to assign opens

## Deactivate or activate a Data Connection

To stop a data flow between two Nodes, you are able to deactivate a data connection.

Of course you are also able to activate connections again

### Deactivate

- **Right-Click** with your mouse on a connection line
- The **Context Menu** opens
- select "**Deactivate...**"

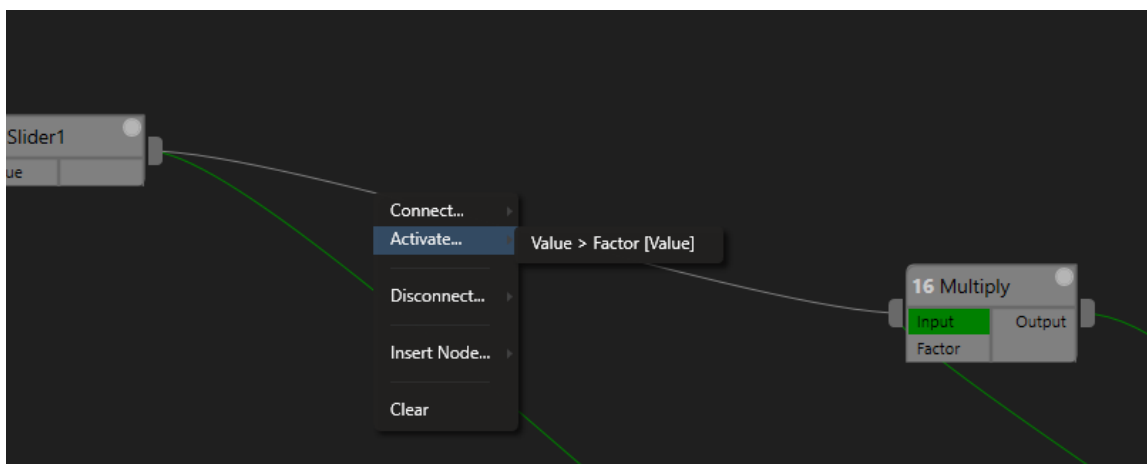


#### **Deactivated Connections are gray instead of green**

You can recognize deactivated data connections by the gray color of the connection line.

If the connection is deactivated, no data flows between the two Nodes.

### Activate



- **Right-Click** with your mouse on a deactivated an gray connection line
- The **Context Menu** opens
- Select "**Activate..**"
- Select the Connection you want to activate



## Composite Node

VERTEX offers you the possibility to **build Nodes that contains a logic of other Nodes**.

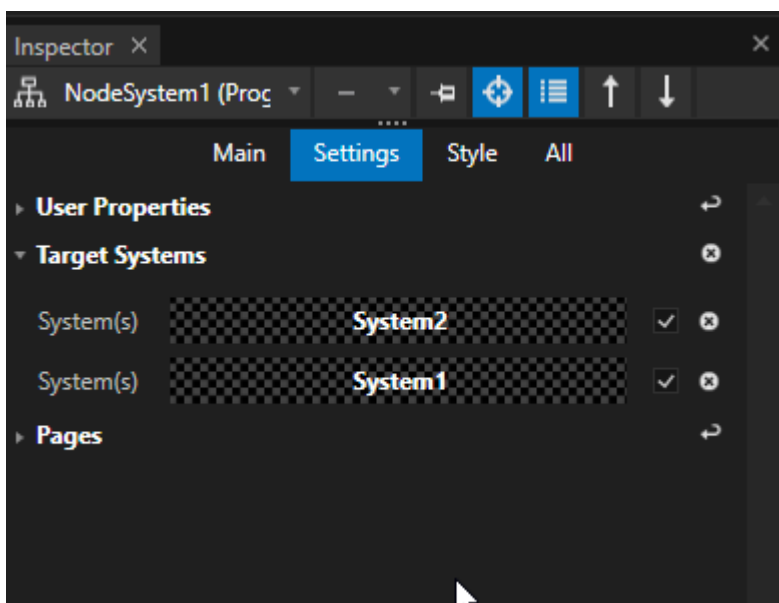
We call this kind of Nodes "Composite Nodes". You are able to built your own nodes as a subcomposition of other nodes. You can work them like normal Nodes and use them in your Node System.

Read more about here: [Composite Node](#)

## Advanced: Target Systems for Node Systems

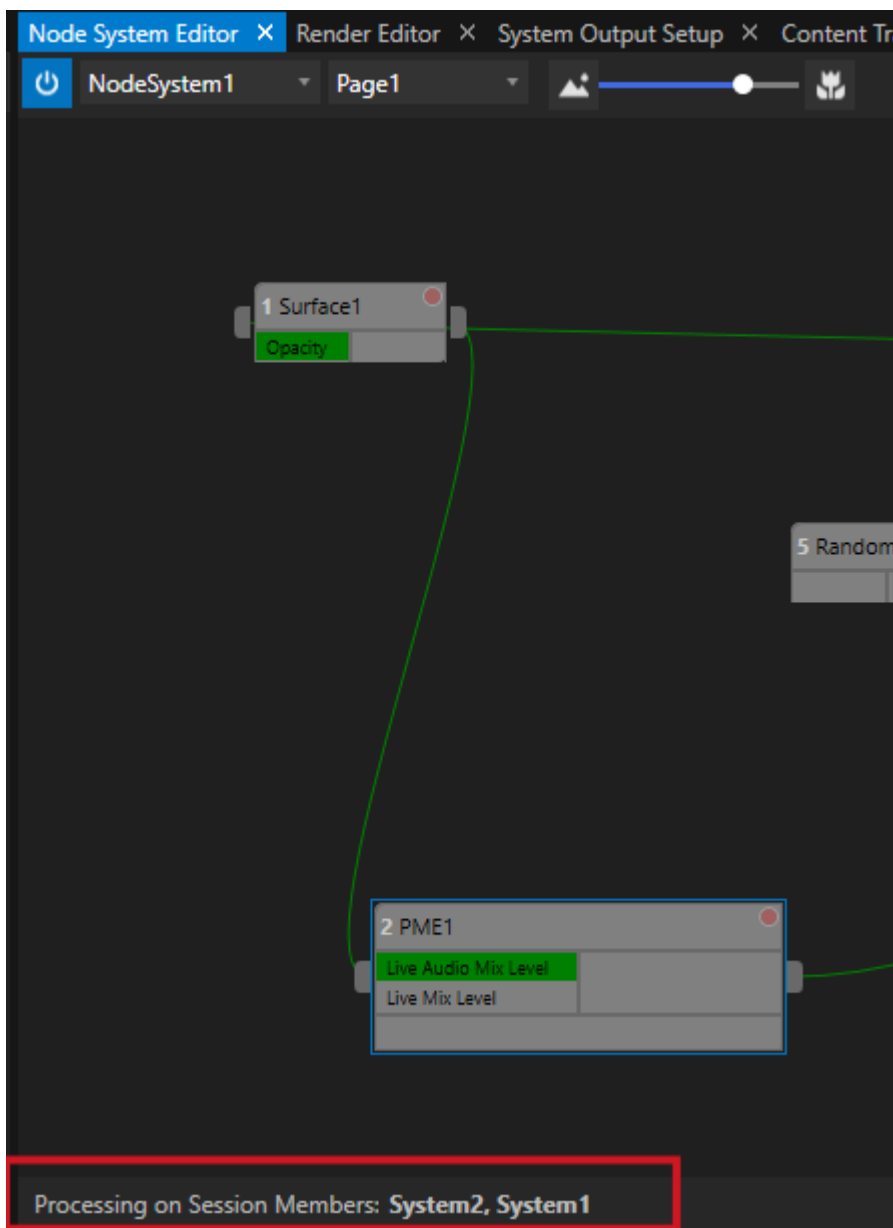
- If working with **multiple Systems into a Project**, you can define **on which of this Systems a Node System should be executed**
- **By default**, a Node System first is only executed and running **on Master**
- If a Node system should run on multiple Systems into your Project, you can define this with the **Target Systems Tab**
- The **Node System Editor** displays on which Systems your Node System is currently executed and running

### Define Target Systems



- First switch to the [Advanced Mode of The Inspector](#)
- Select a Node system into Project Explorer
- Open the "Target Systems" tab into the Inspector
- Drag a System from the Project Explorer to the Target Systems Tab
- Repeat the last step for all Systems on which you want to run the Node System

There is an information into Node Systems Editor on which Systems of your Project your Node System is currently processed:



## 6.9.5 Parameters and Conditions

- Most Nodes have **Parameters** where the data source is defined: This could be a **Connected Input**, A **Literal value** or a **Property Value**
- Some Nodes are working with **one or multiple conditions** that could be true or false
- There are **different options to validate conditions**

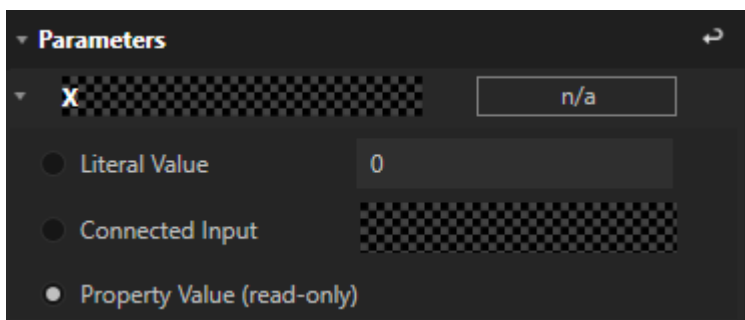
### Parameters

Most of the Nodes are working with Parameter(s) for data. Depending on the type of the Node, there are different options.

**A Parameter could be:**

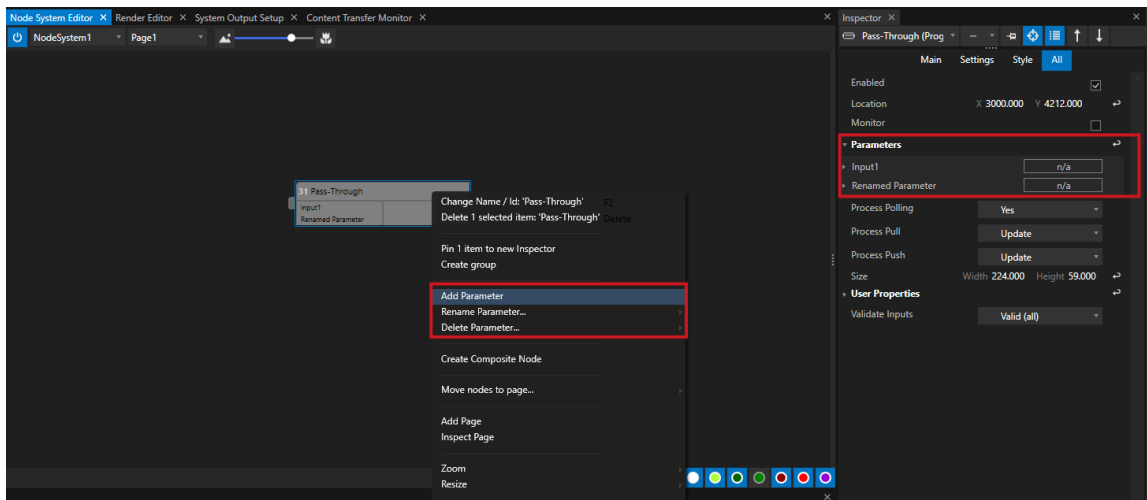
- A **Connected Input** that is connected to an output of another Node and receives data
- A **Literal Value:** a fix value that is manually added by you
- A **Property Value:** a Property Value of an item. This could e.g. be an Input Value of a Device, a Dimension x-coordinate of a Canvas, an Opacity of a Surface, a Position of a Clip Container,...

For every item into VERTEX, properties are listed into the Inspector. Most of them (not all) could be used as Parameter Inputs of a Node.



### Add a Parameter

In most cases during your work with Nodes, Nodes already have parameters after they have been created. Furthermore, there is an option to manually add another Parameter.



- **Right-Click** on a Node
- The Context Menu opens
- Select **"Add Parameter"**
- For some types of Nodes, you can select between "Text", "Float", "Integer" and "Boolean"
- A new Parameter is added to the selected Node



#### **A New parameter is not displayed in a Node or not shown in the Inspector?**

Please enlarge the node downwards by dragging the lower edge of the node with the mouse. The new parameter should now be visible.

If the new Parameter is not displayed into the Inspector, please refresh the Inspector by selecting the same Node again.

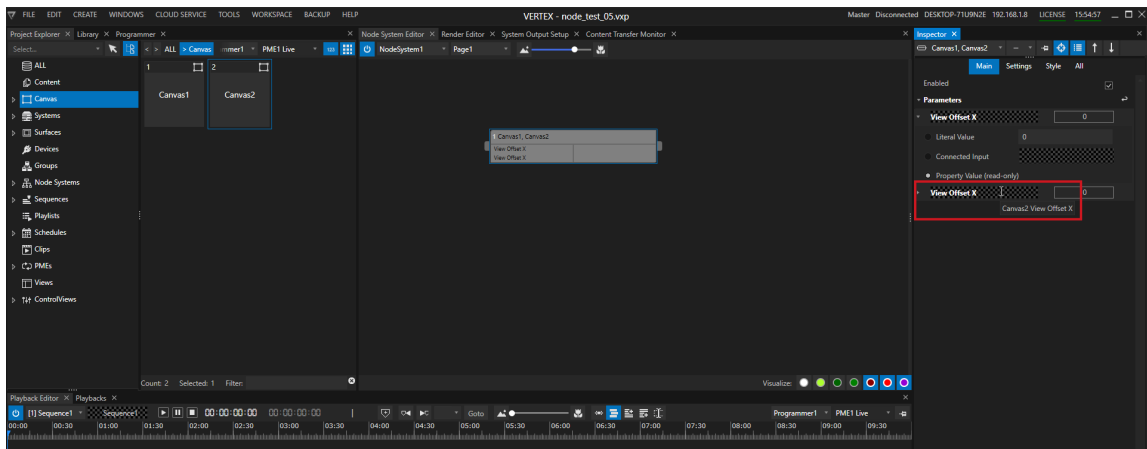
## Rename a Parameter

- For some Nodes (e.g. the Pass-Through Node) it is possible to rename a Parameter.
- Click on the Node, open the Context Menu and select "Rename"

## Delete a Parameter

- Right-Click on a Node
- The Context Menu opens
- Select "Delete Parameter"

## Show full Parameter source



### Show full property source of a Node Parameter

For a better and cleaner overview - the property fields of a Node in the Inspector show reduced information: They only show the property name but not the full origin. When working with combined Nodes, a **tooltip displays you the whole property source**. To show this tooltip, **hover with your mouse over the parameter field into Inspector** or over the parameter field of the Node into Node Systems Editor.

## Advanced: Input Mode for Property Parameters

- There are special and advanced settings for Parameters that have a Property from a Project Item (e.g. a Clip Container, a System, a Surface) as source
- You are able to define a **mode how the property modulation from a node will relate to data for this property** that is used and modified on other places into VERTEX

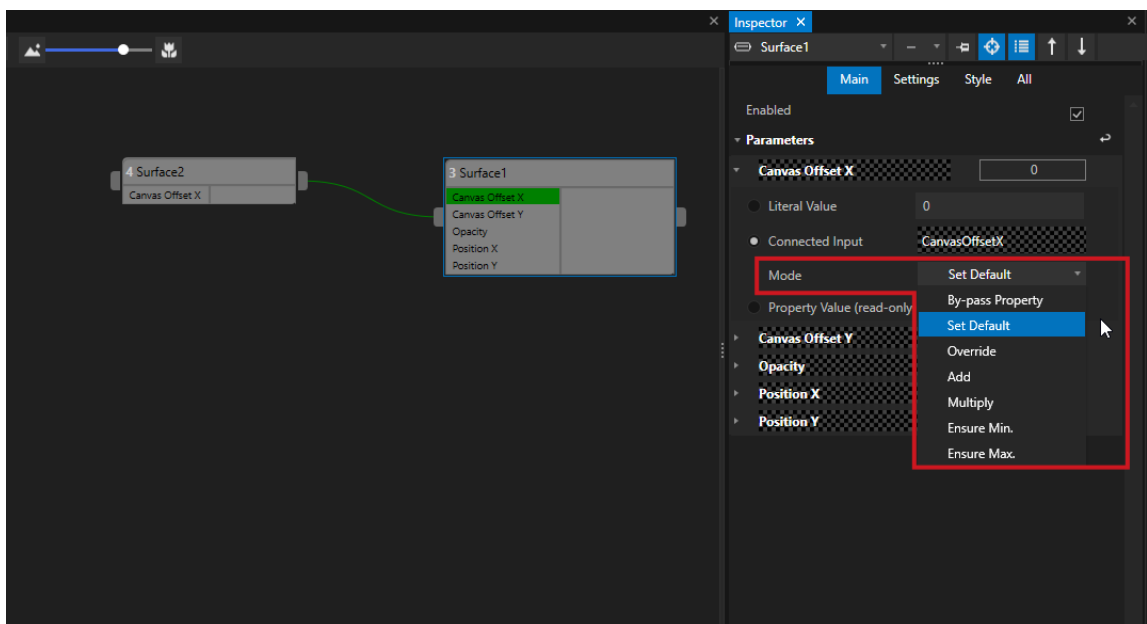


### Data Modulation for Properties and the processing chain in VERTEX

In VERTEX there are many different ways for you to model the data of a property (e.g. the opacity of a surface). Manually with the Fader in the Inspector, but also with e.g. Wiring, Triggering or with values, which are written into the Programmer.

In VERTEX internally all these influences are processed on a value. Because it is software and complex math this must happen after a certain order. Node Systems are the last part of such an internal processing chain under the hood of VERTEX.

The result of a node system - a **so-called Modulator** - is considered and added last, before a property value is finally rendered or output.



#### **Bypass:**

The property will not be affected. Consuming Nodes will use the input's value, regardless of the property's value.

#### **Set Default:**

The default value of the property will be changed according to the Node's Input value.

#### **Override:**

The modulator for the property will override the previously evaluated value with the input value of this Node

#### **Add:**

The modulator for the property will add the input value of this Node to the previously evaluated value

#### **Multiply:**

The modulator for the property will multiply the input's value with the previously evaluated value.

#### **EnsureMax:**

The modulator for the property will ensure that the resulting value is not greater than the input's value.

**EnsureMin:**

The modulator for the property will ensure that the resulting value is not lesser than the input's value.

# Conditions

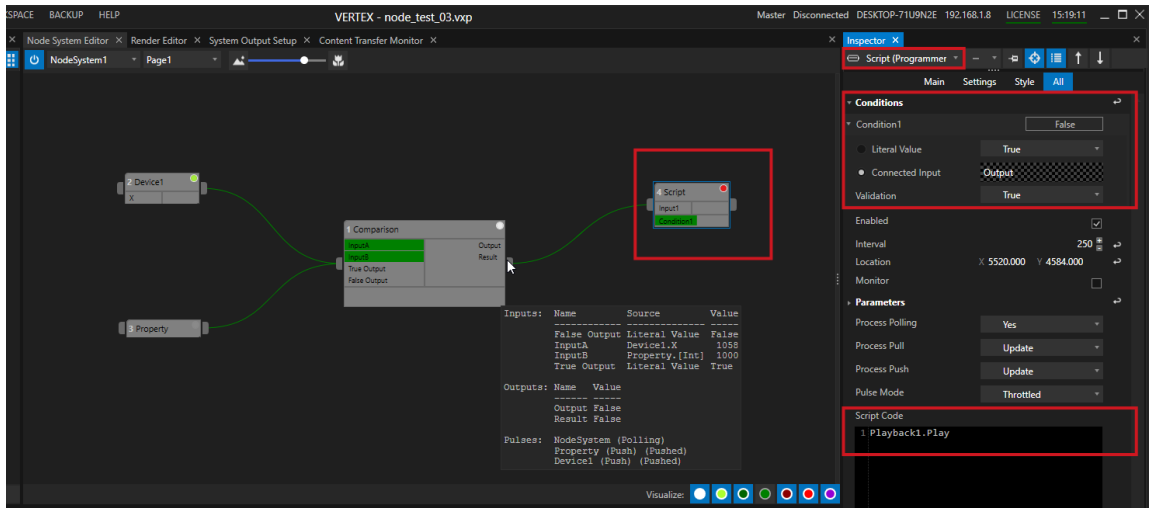
- A condition can be **true or false**
- When the selected validation applies, a **Script is triggered**
- You can **choose from a list of validation options** - the default validation is true ( if the input is "true", a Script is triggered)

i

**Accepted Input Values**

Nodes with a Condition expect only values at the input that are either true or false or the equivalent numbers 1 and 0.

Other numbers and other values are therefore not accepted.



Example for a Condition:

A "Compare" Nodes compares two incoming Values. If Input 1 is bigger than Input 2 it returns a "False". This logical result is connected to a Script Node that expects a "True" or a "False" as Condition. Because "False" is the incoming data, the Script Code is not executed. The "Visualize Pulse" Monitor shows a red dot.

## How to do

- Create a Node that works with a Condition (e.g. a Script Node)
- Or add a condition to an already existing Node (use the context menu with a right click)

- Connect the Node with a valid data connection - the incoming data from another Node has to be a Boolean "True" or "False"
- Select the Node into Inspector
- Fold out the "Conditions" Section there - check the options for validation
- Go to Inspectors Main Tab
- Enter some Script Code that should be executed when the validation is reached



### **Change to Advanced-Mode to see added Conditions for some Nodes**

Basically you are allowed to add conditions to almost all type of Nodes - also those that a first created with only Parameters.

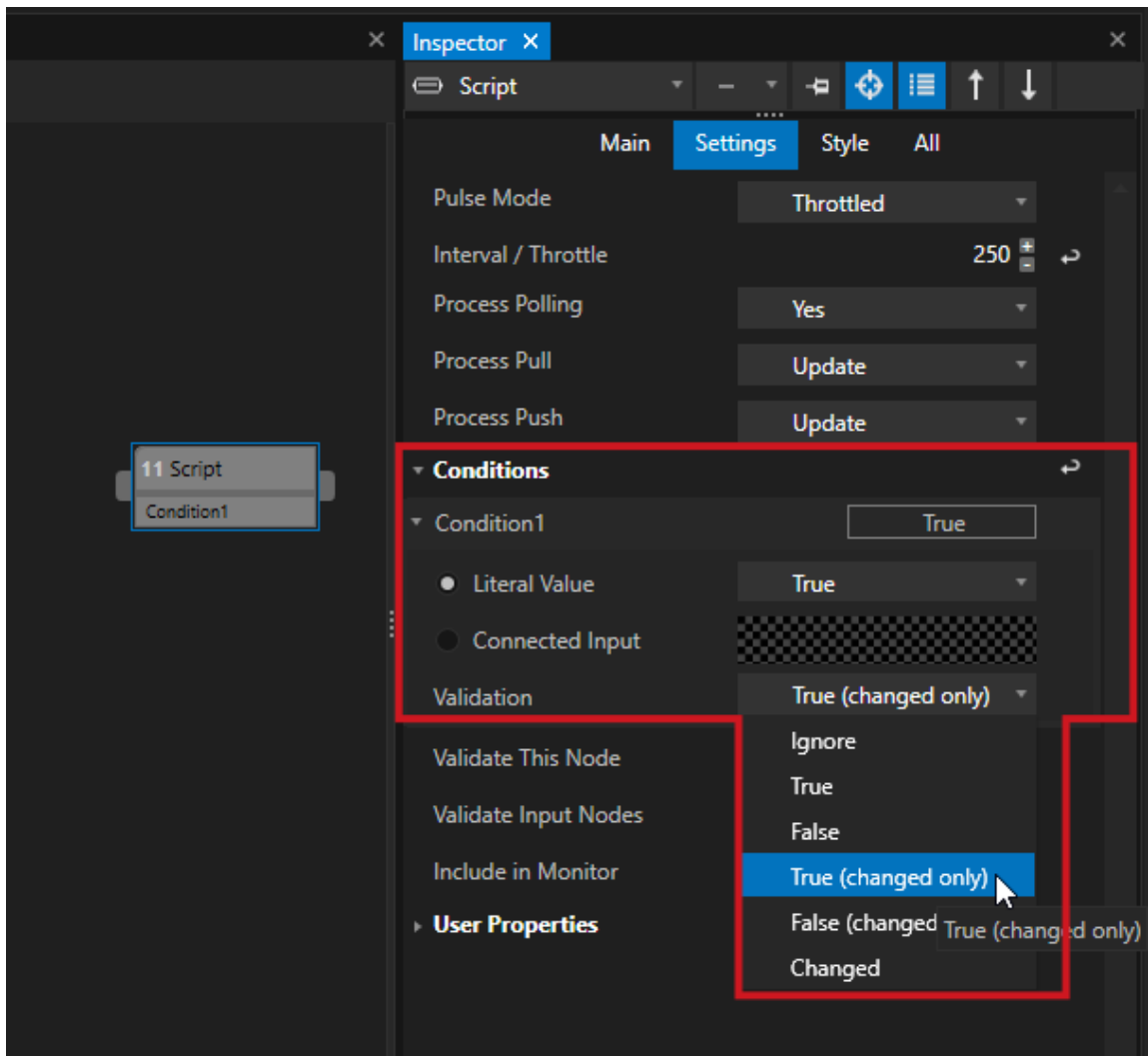
If "Add Conditions" is offered into the Context Menu of a Node (right-click with your Mouse on a Node), it is possible to add a Condition to this Node. Depending on the Node type it could be that you have to change the [Inspector Mode to Advanced](#) to see the Condition for your node in the inspector besides the Parameters

## Validation

You are able to change the validation type of a Condition.

Just expand the menu from a Condition into the Inspector.





There is a dropdown where you can set one of the following validation options:

**Ignore:**

Ignore incoming Data

**True:**

Execute Script if Incoming data has the Boolean value "True"

**False:**

Execute Script if Incoming data has the Boolean value "False"

**True (Changed only):**

Execute Script if Incoming data has the Boolean value "True" and has changed from "False"

**False (Changed only):**

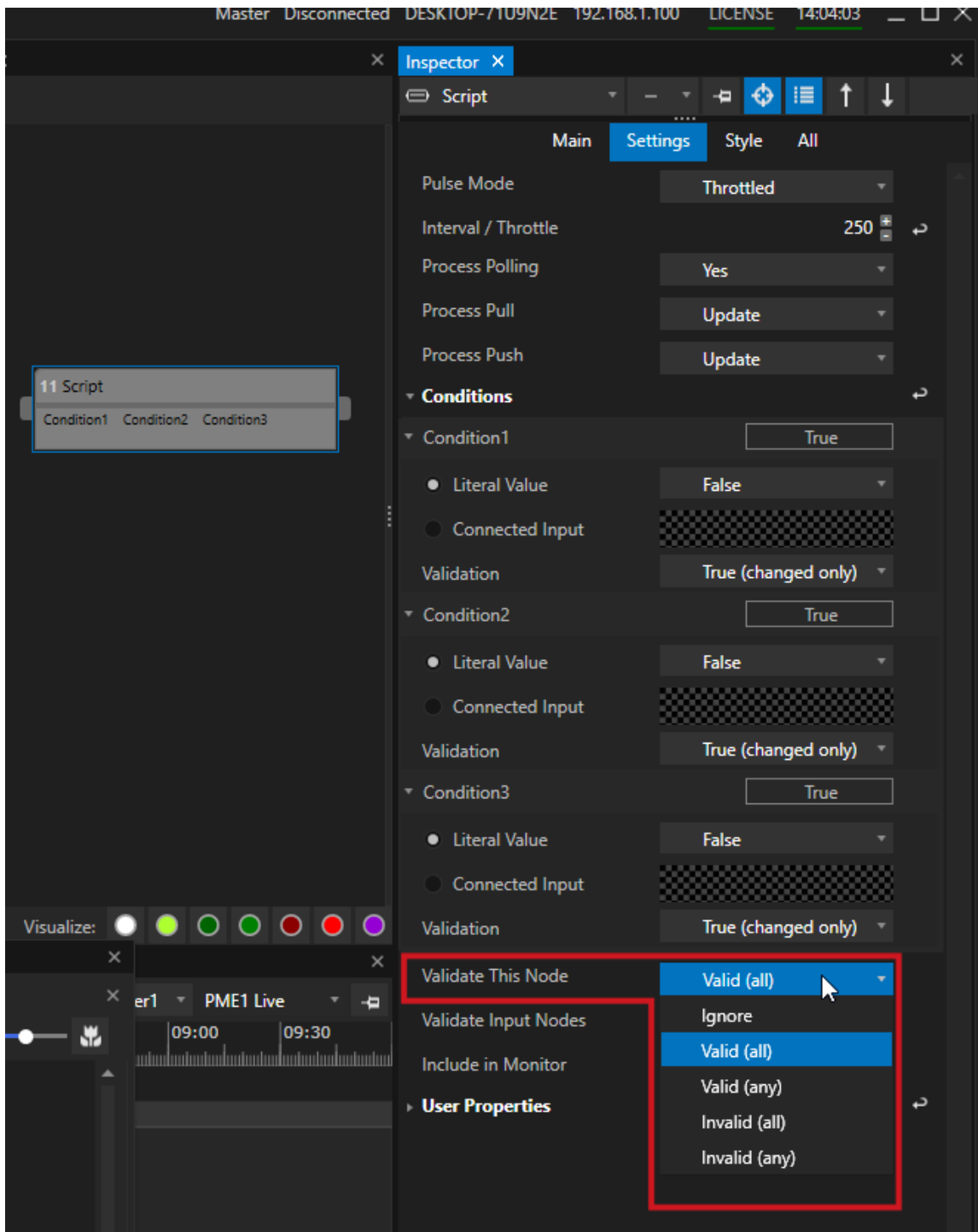
Execute Script if Incoming data has the Boolean value "False" and has changed from "True"

**Changed:**

Execute Script is incoming Boolean has changed its value

## Validate This Node

- If working **with multiple Conditions in the same Node**, "Validate this Node" gives you the option to **make the Node behavior dependent on several incoming conditions**



A Script Node with 3 Conditions as Input

**Ignore:**

Ignore the validation of all conditions

**Valid (all):**

*Node is only executed if all of the conditions are valid*

**Valid (any):**

*Node is only executed if in minimum one of the conditions is valid*

**Invalid (all):**

*Node is only executed if all of the conditions are invalid*

**Invalid (any):**

*Node is only executed if in minimum one of the conditions is invalid*

## Advanced: Validate Input Nodes

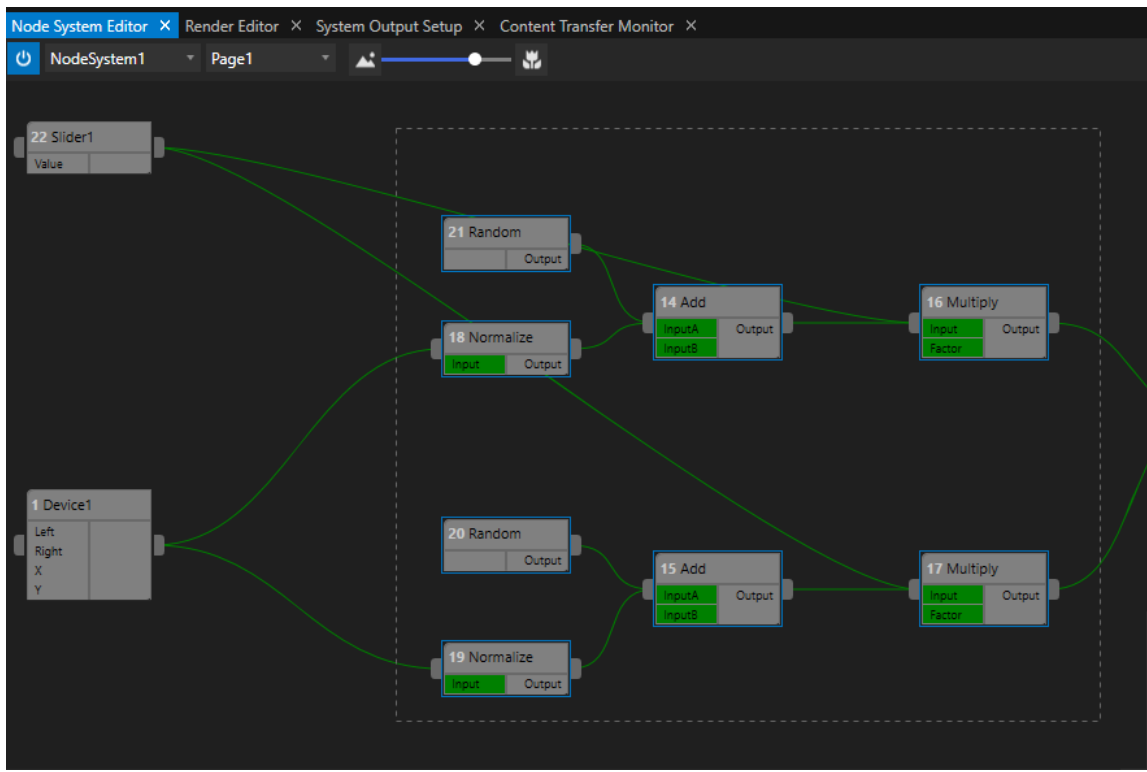
- To control whether nodes are processed in order to conserve performance, nodes can be "validated", i.e. they are only executed if certain conditions are met.
- These can be own conditions ([Validate this Node](#)) or conditions of nodes connected as input, which they "inherit".
- "Validate Input Nodes" ensures that all Nodes or a chain of Nodes that are connected to this Node are valid or even not.
- If the validation fails, the Node will not be processed

### 6.9.6 Composite Node

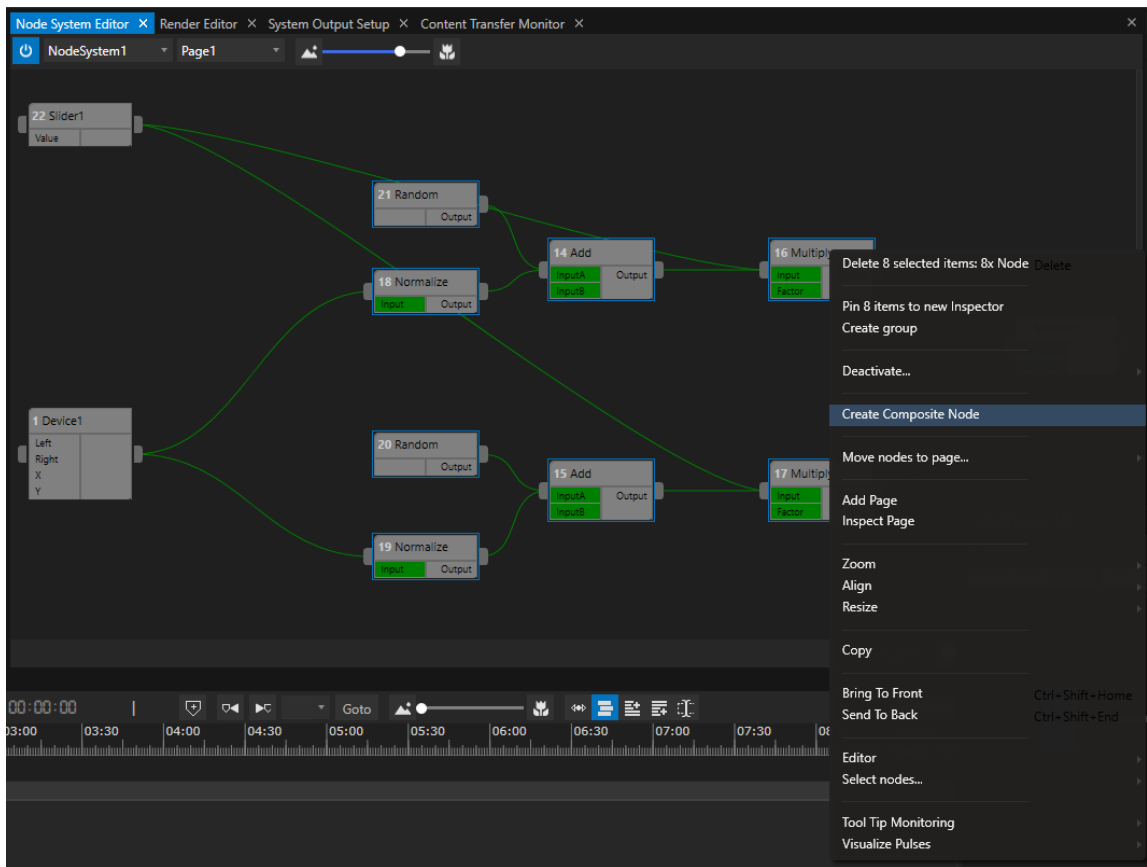
- With the **Composite Node** feature of VERTEX you can **combine an arrangement of Nodes to only one single Node**.
- You can then work with this **sub-composition as just one Node**.
- You are able to **decompose** a Composite-Node again.

#### Create a Composite Node

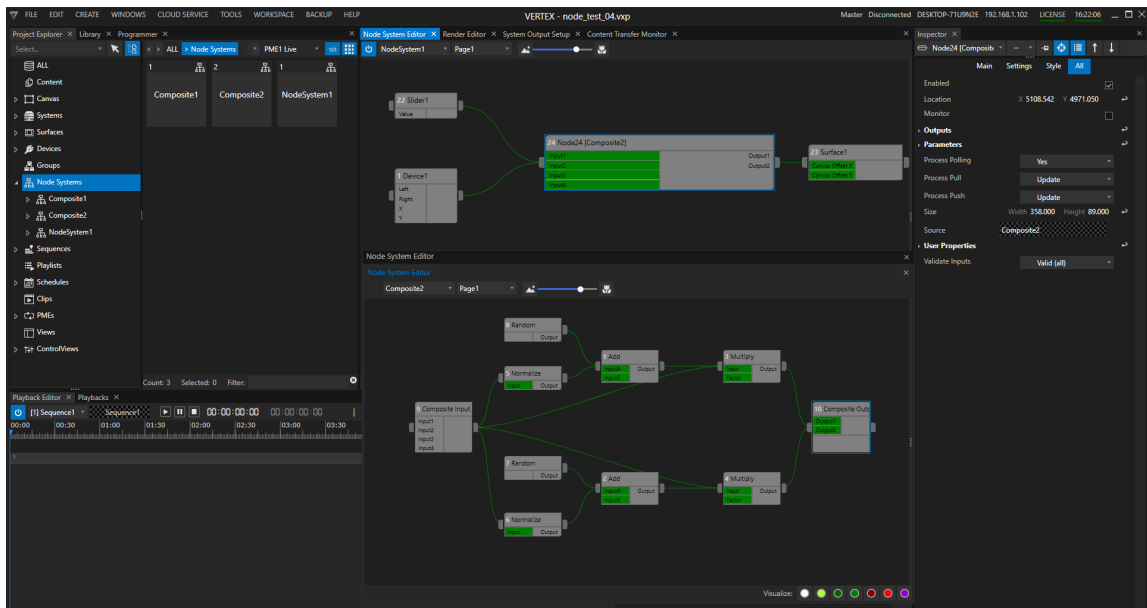
- **Hold your left mouse button and draw with your mouse a selection** around the nodes you want the Composite to contain
  - **Release the mouse button:** the selected nodes now are outlined in blue
- or
- **Hold the CTRL- Key, select single Nodes to a multi-selection**



- **Right-Click with your mouse** inside the Node System Editor
- The **context menu** opens
- Select **"Create Composite Node"** there



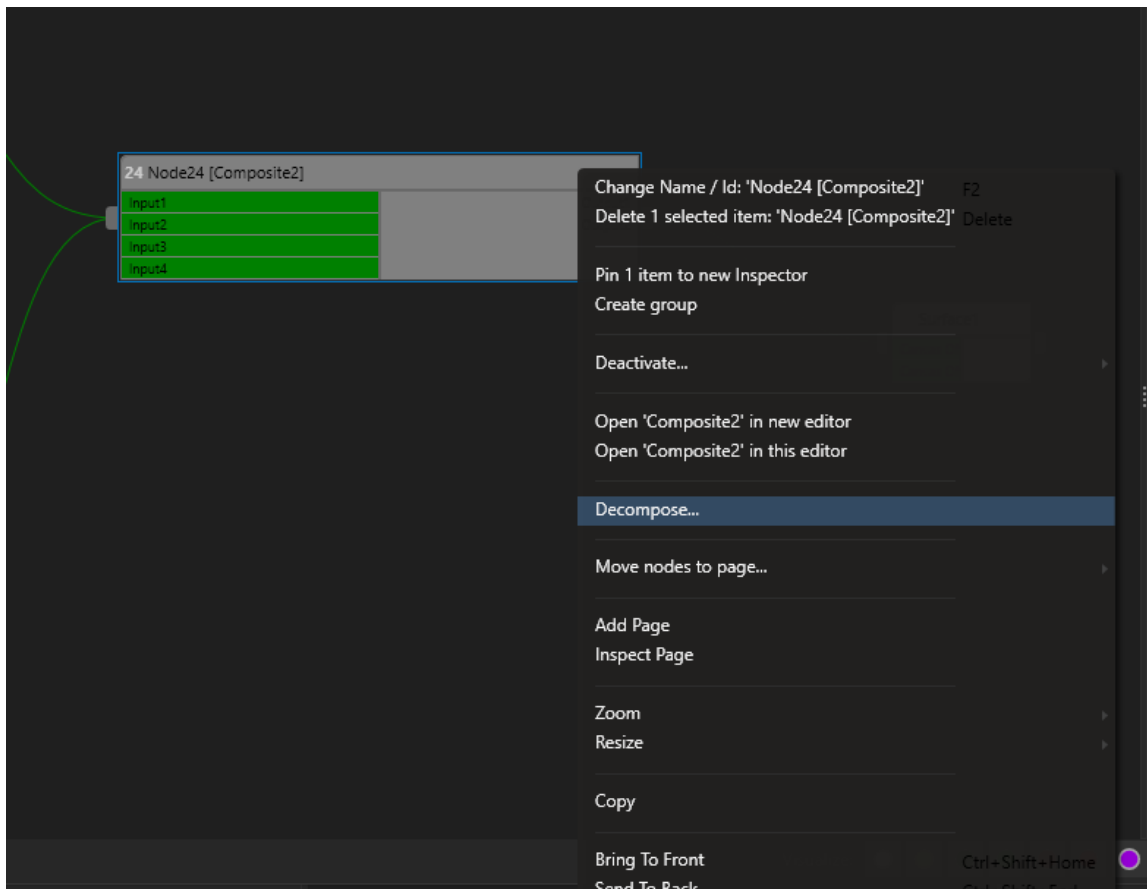
- A new Composite Node is created. The **input and output Parameters** of this Composite are **automatically created** out of the Node selection
- You can display the Composite and all of its child Nodes over the **Node System selection dropdown** of the Node System Editor
- You also are able to **access the Composite** and its child Nodes **over the Project Explorer**. If you want to use the Composite Node twice, just **drag it from Project Explorer to a Node System Editor**



**i** **No Undo for Composites**  
 There is no Undo possible (with e.g. CTRL+Z) for Composite. If created accidentally a wrong Composite, you are able to decompose it (see below)

## Decompose

- Select the Composite Node into Node System Editor
- Right-Click on it to open the Context Menu
- Select "Decompose"
- Confirm with OK



### **Composites live on**

*If you do a decompose of one of your Composites this only will be done into your current Node System.*

*The Composite still exists into Project Explorer until you will delete it there. As long as it is alive, it can also be selected via the dropdown list in the Node System Explorer.*

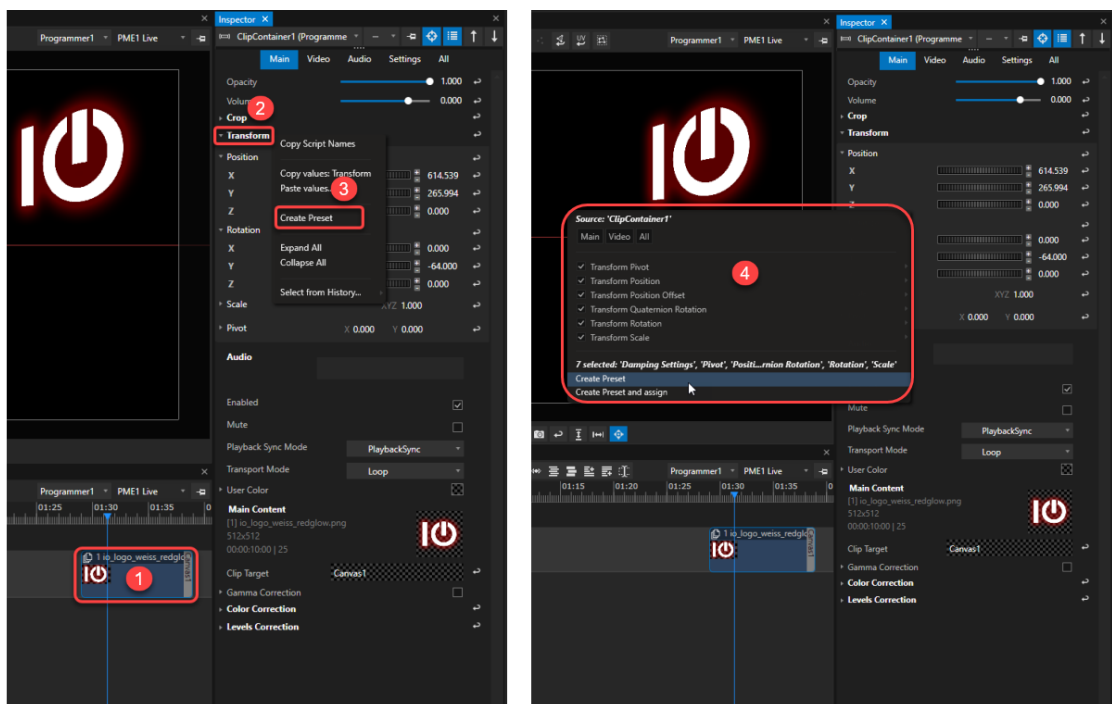


## 6.10 Presets

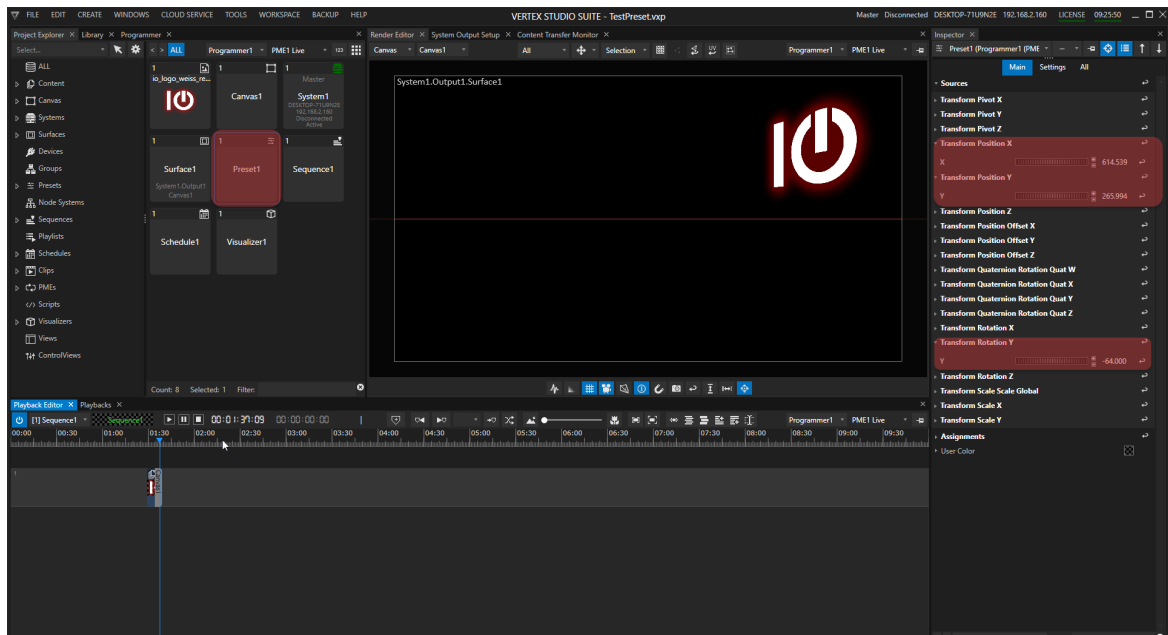
- VERTEX offers a quick and efficient way to **store multiple inspector settings as a whole in presets**.
- Presets can be **assigned to multiple project items with one click** - a real time-saver when preparing lots of content for your show.
- At the time of VERTEX R3 in 2022, **all properties that are controllable by animation keyframes can be stored as a preset**.

### Creating a Preset

Let's say, you have an image on your canvas in a certain position that you would really like to use more often. To store its Transform properties in a preset, just follow the next four steps:



1. Select the item and access its properties in the inspector.
2. Right-click on the Transform properties to open the context menu and...
3. ...select Create Preset
4. A new dialogue window will ask you to select what particular properties you want to include in your new preset.



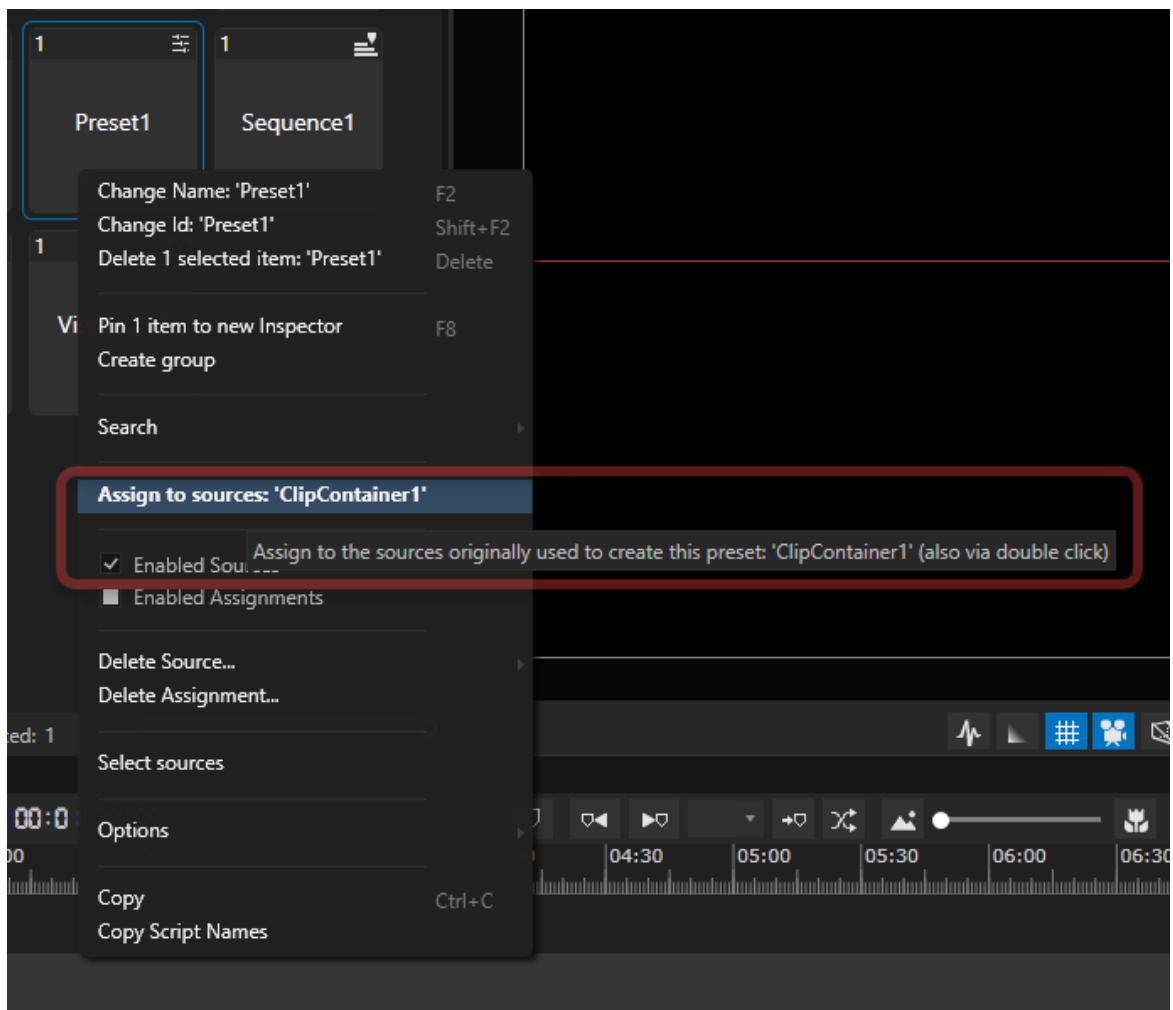
*A new preset has been created. If you select it in the project explorer, the inspector window will show all the source properties that have been stored in the new preset.*

## Assigning Source & Destination

*Now you have created a new preset from the transform properties of clip container 1.*

*If you change those properties in the clip container for some reason and need to go back to what you stored as a preset, you can assign the preset back to its source:*

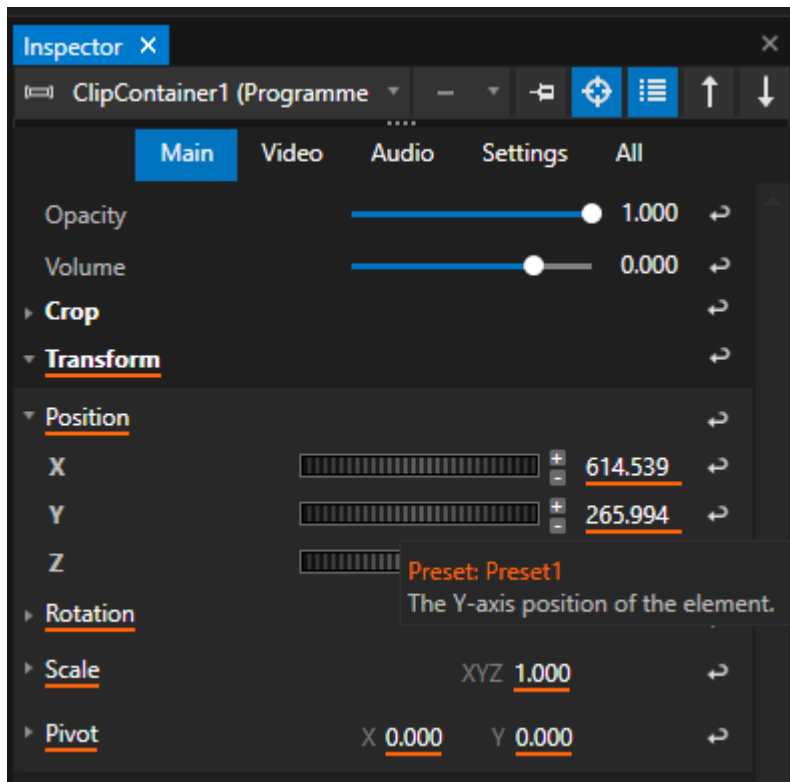
*Either by double clicking on the preset or via the right-click context menu:*



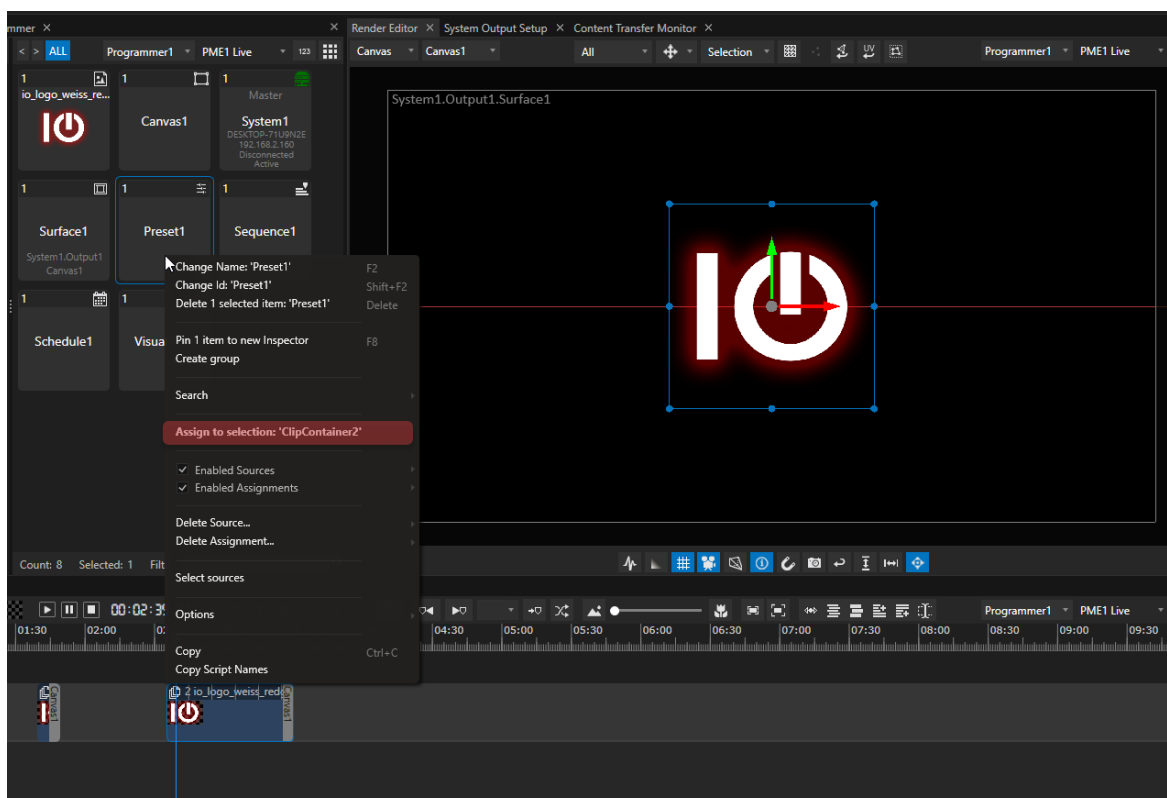
There's also a third option to assign the preset. Just drag it from the project explorer and drop it into the clip container you want to assign it to.

At any rate, if you take a look at the inspector window, you will notice how the Transform properties of Clip Container 1 are now underlined in orange.

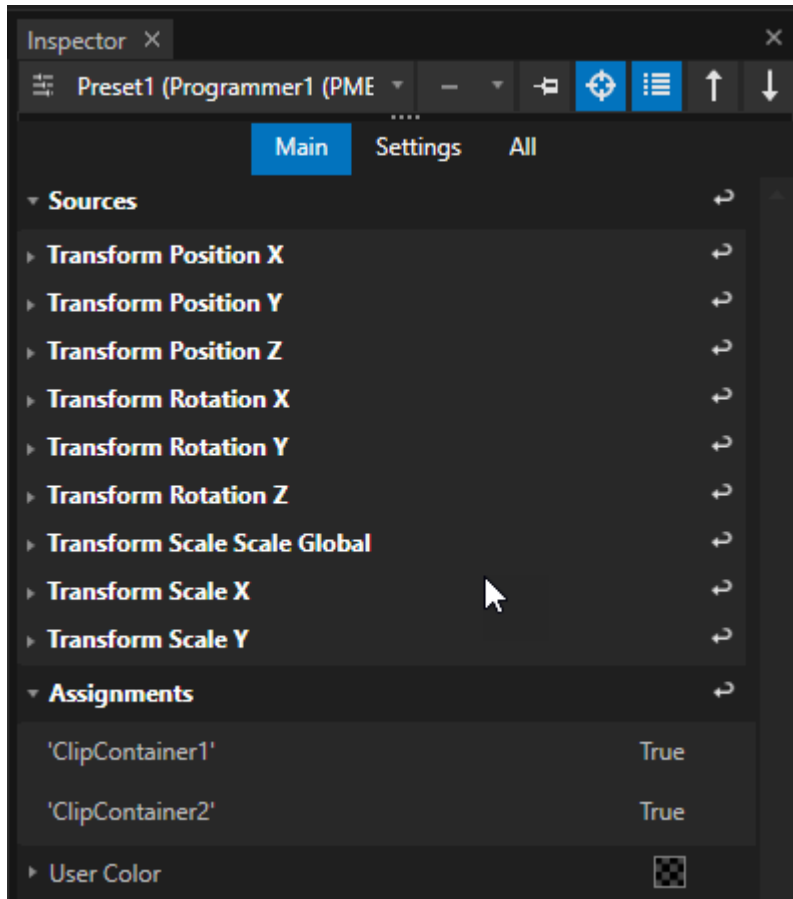
This is to indicate that the property values now come from a preset and no longer from the operator in Value Mode.



Assigning the preset to a different clip container works like the same way mentioned above:  
 Select the clip container in your sequence and access the preset's context menu with a right-click where you choose Assign to selection:

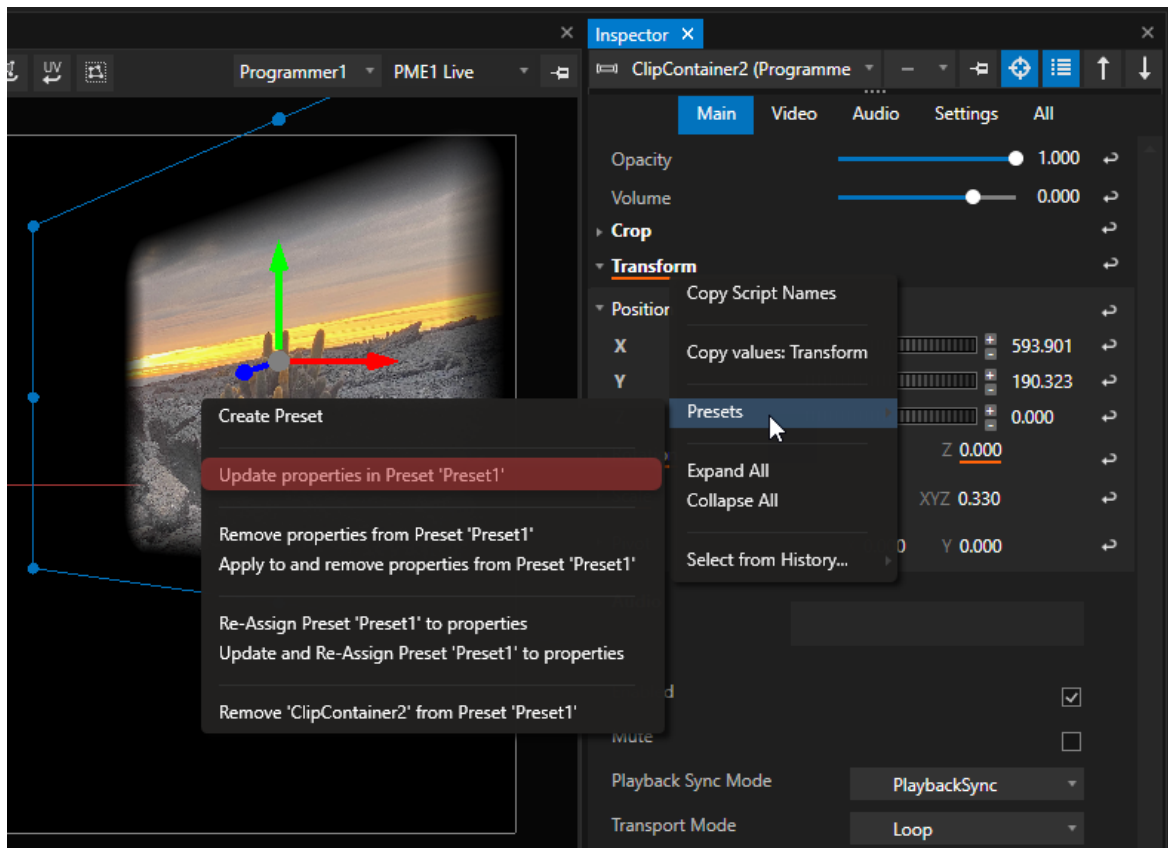


Notice how the preset's Assignments list in the inspector has grown by one item.



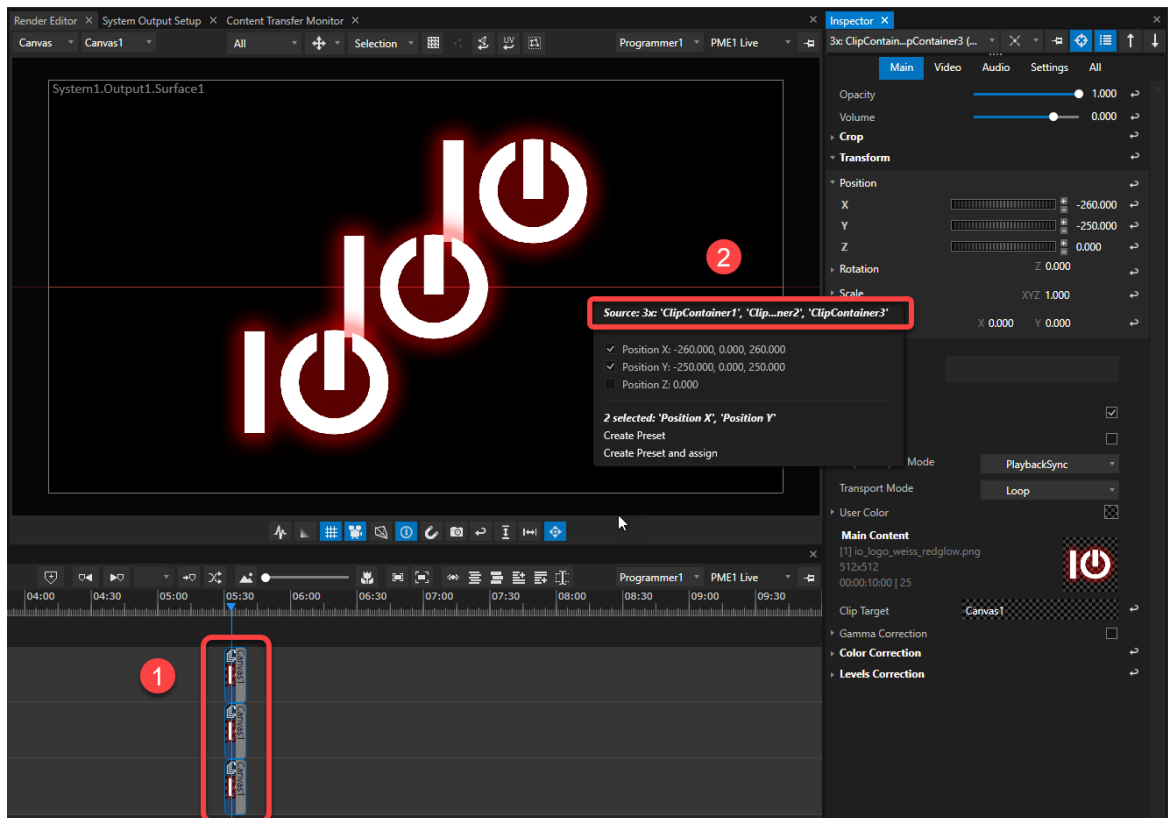
## Updating a Preset

If you need to change the property values stored in a preset, simply access the Presets context menu in the inspector by right-clicking on the values that have been changed and choose *Update properties in Preset* from there:



## Group Presets

*A preset can not only store properties for one single item, but also for multiple items in a group:*



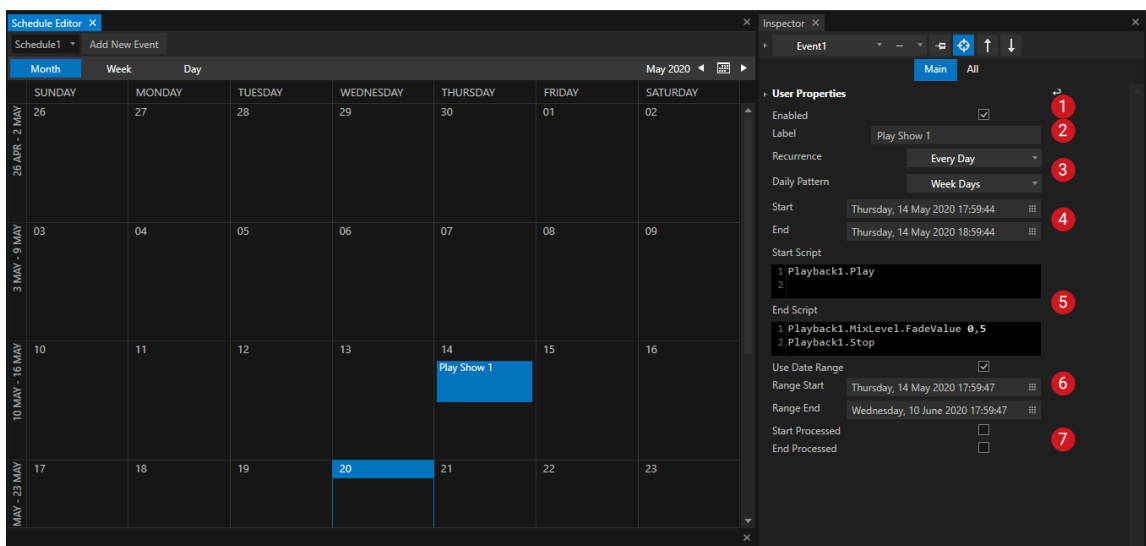
- 1) To create a group preset first select a group of items (clip containers, DMX devices etc.)
- 2) Access the context menu of a property in the inspector. The preset source will now name a group of items (3 clip containers in the example above).

Position and offset of this group can now be assigned to any other group of items. Keep in mind, that **if the assigned group is larger than the source**, VERTEX will automatically **stop interpolating at the last or highest value**.

## 6.11 Schedule Editor

- The **Schedule Editor** provides you with the option to **execute script commands by a specified time and date**.
- The Schedule editor comes with a **calendar based user interface**.
- Access all settings in the inspector.

### User Interface



<b>1</b>	<b>Enabled</b>	Enable or disable an event.
<b>2</b>	<b>Label</b>	Add a custom label for your event. This Label is also displayed as the event's name in calendar view.
<b>3</b>	<b>Recurrence Daily Pattern</b>	Sets the Recurrence. Default: Once - event will be triggered only once Option: Recurrence from Minute to year



4	<b>Start and End Time of the Event</b>	Sets the start time of your event and the end time. At start and at the end separate scripts can be executed
5	<b>Script Commands</b>	<b>Start Script</b> Enter a Script or single Script Commands that should be executed at the start of your event <b>End Script:</b> Enter a Script or single Script Commands that should be executed at the end of your event
6	<b>Date Range</b>	<b>Use Date Range</b> Default: Disabled Enable to use only a time span in which the event will take place. <b>Range Start</b> Start date for the time span. First date on which the event is executed <b>Range End</b> End date for the time span. Last date on which the event is executed
7	<b>Start Processed and End Processed</b>	Information on if and when an event was executed. The check is set by vertex when the event start has been executed. Useful to supervise if the event start has taken place, when working with a recurring event. <b>Start Processed</b> Will be checked when event start is processed <b>End Processed</b> Will be checked when event end is processed

## Create a Schedule with an Event

1. Go to the **"Create" Tab into the Main Menu** and **create a Schedule**  
A first Schedule is already created by VERTEX when you start with a new project
2. Go to the **"Windows" tab into Main Menu** and open a new Schedule Editor
3. **Select a Schedule you want to edit** in the dropdown of the Schedule Editor
4. **Select a date** into calendar.
5. Click to the **"Add new Event" button**
6. Select the new Event in the calendar and adjust settings in the Inspector.

## 6.12 Scripting

- VERTEX comes with its own script language which is super easy to learn.
- **Many places in VERTEX have a script command field** (cues, systems, clip containers, etc.) from which you can **control virtually anything with a script command**.
- You can **combine multiple script commands into one script**.

### Steps to start with Scripting in VERTEX

#### VERTEX Scripting

Learn how VERTEX script commands are structured and how their basic syntax works.

Get to know how to assign values and how to fade to values.

Use tags to jump inside scripts. Learn how to operate local variables.

#### Script Editor and Script Monitor

VERTEX offers you two tools to make scripting as simple as possible.

Use the SCRIPT EDITOR to write, access and edit all scripts your project contains.

The SCRIPT MONITOR shows all detailed information about status, execution and errors of script commands and helps debugging your scripts.

#### Scripts

Scripts may contain multiple script commands.

In one script you can bundle a list of script commands and execute them from top to bottom.

There is an option to define one or more parameters for a script. Values for those parameters will be set on script execution.

#### Variable

Define variables with a value and use them for recurring operations.

Learn more about the difference between global and local variables.

**Show a list of all available Script Commands.**

VERTEX comes with a dynamic script language: its commands follow the same logical process as working with the GUI.

**To show a list of all available script commands**, just click into one of the script fields or the command section in the status bar at the bottom and **press "CTRL and Space"**. A list of all available commands opens. Select an item into list and press ENTER. To show **all available commands for the next deeper level**, enter a decimal point "."

## 6.12.1 Vertex Scripting

- Script Commands are **GUI-oriented** in regards to VERTEX's structure and workflow.
- You can **trigger simple actions, switch settings, or operate with values**.
- Get to know the **basic structure & syntax** and familiarize yourself with some simple script examples.
- To learn how to create a script [go to the topic Scripts](#).

## Simple Actions

Enter short and simple commands into a scripting field or into the command line and trigger an action.

```
Playback1.Play  
Playback2.Stop  
Playback3.GotoCue 1
```

## GUI-Oriented Structure

Script Commands are **tailored to resemble working with VERTEX's graphical user interface**.

The syntax and structure should look very familiar to you if you already know how to work with the GUI.

Almost every property in VERTEX can be scripted.

## Example: Set Opacity of ClipContainer1 in Playback1 to a Value



Follow the numbered markers in the screenshots to understand the order in which script commands are composed and what structure in the GUI they represent.

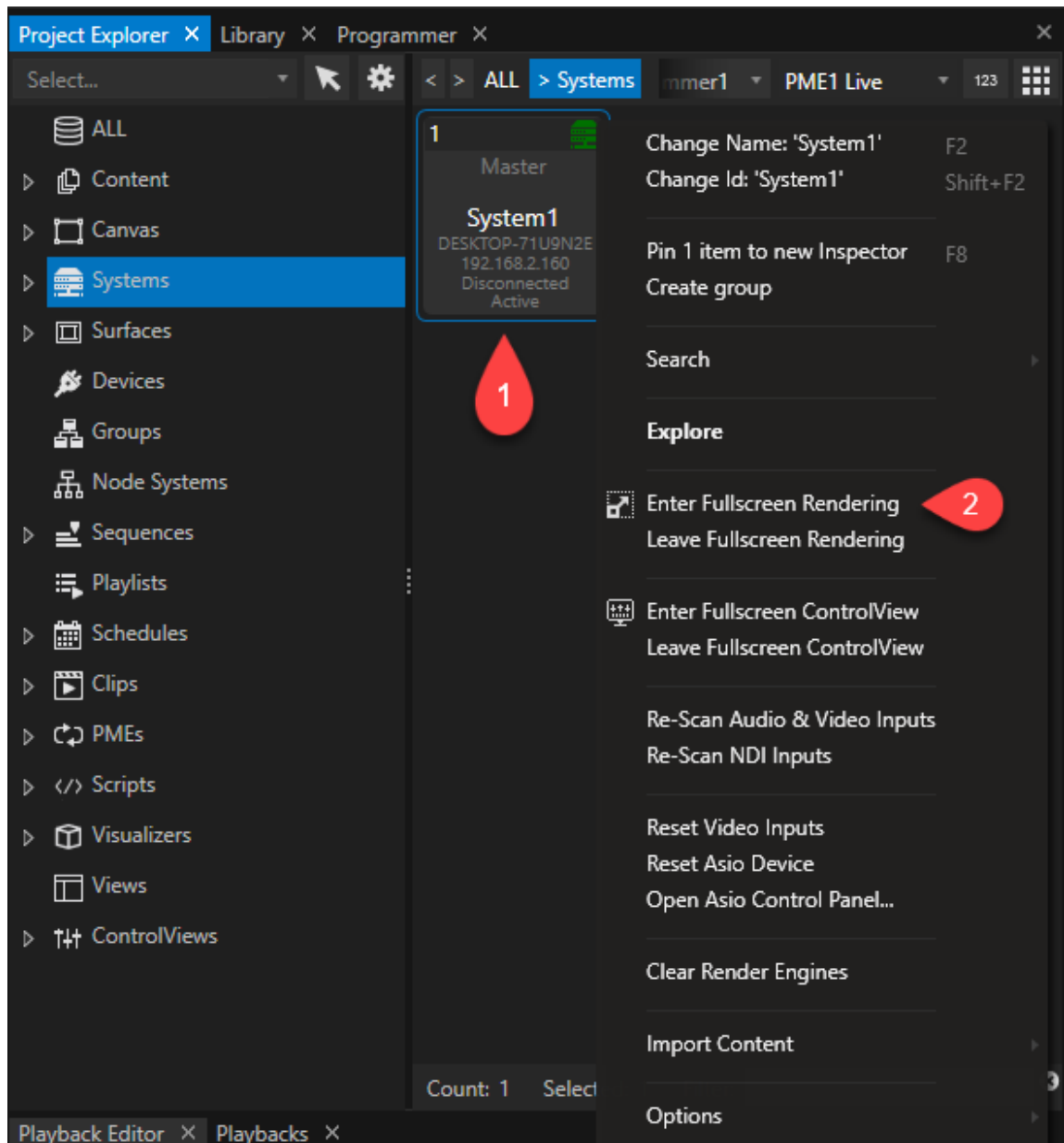
For example: In Sequence1, ClipContainer1 set opacity value to 0.5:

1 + 2 + 3

Script Command:

```
Sequence1.ClipContainer1.Opacity.Value = 0.5
```

## Set System 1 to Fullscreen



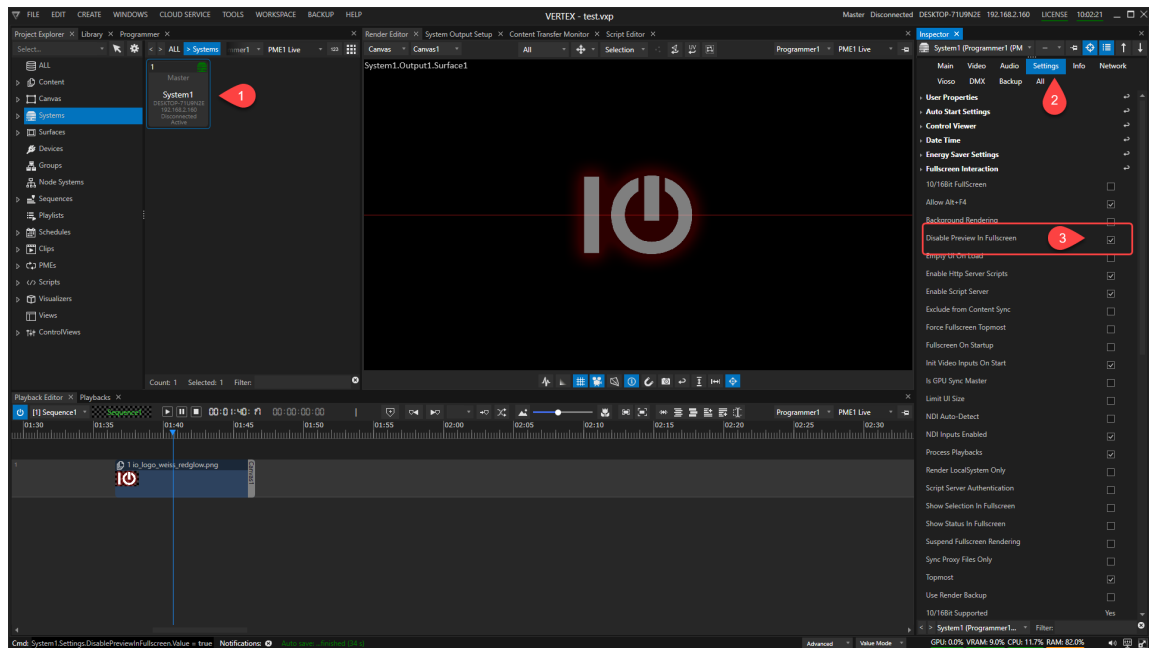
Set System 1 to Fullscreen



Script Command:

```
System1.EnterFullScreen
```

## Disable Preview in Fullscreen for System1



Disable Preview in Fullscreen for System1



Script Command:

```
System1.Settings.DisablePreviewInFullscreen.Value = true
```

## Autocomplete ScriptWizard

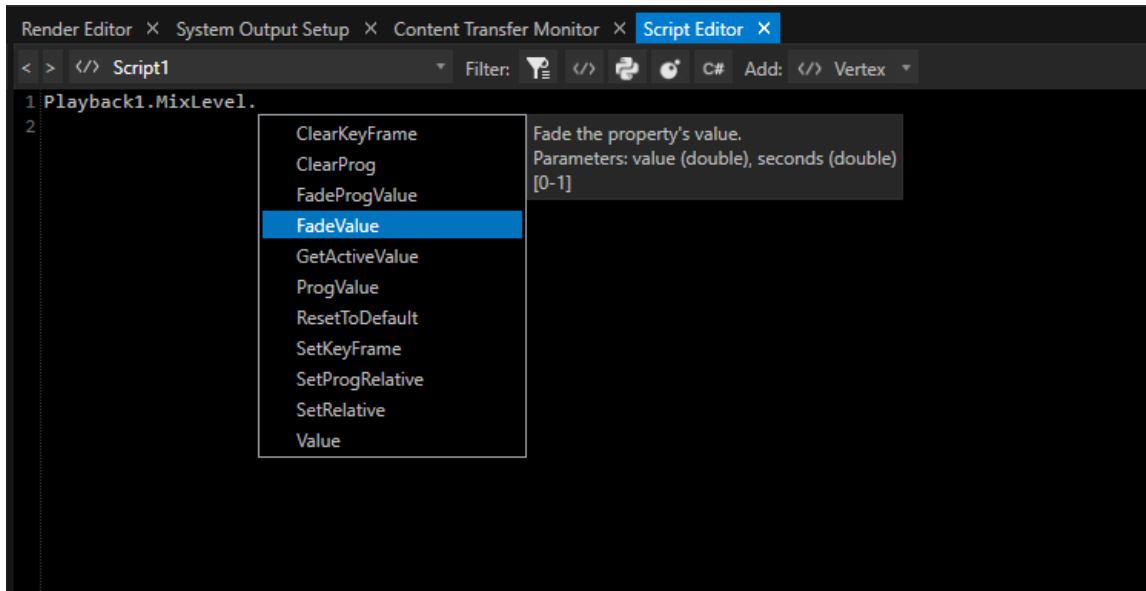
In every scripting field VERTEX's ScriptWizard will provide you with a list of all available items, commands and actions.

Use the shortcut "**CTRL+ Space**" in an empty line to open a full list of all available elements.

**Enter a decimal point** to open a list of all available child options and elements.

For every list item the **expected parameters** are shown.

*Confirm* your choice with **ENTER** for the created script to take effect.



## Structure and Syntax



*The availability of script commands and syntax options depends on your current VERTEX version.*

*With every version, we improve and extend the software feature set - including script commands.*

*Not all script commands or options like local variables or script tags are supported by older versions.*

### Basic structure

*The next deeper level is initiated by a decimal point.*

```
Firstlevel.Secondlevel.Thirdlevel.
```

*One script command per line, use a new line for the next script command .*

```
Playback1.Play
System1.Settings.DisablePreviewInFullscreen.Value = false
```

Comments are preceded by a double slash.

```
//This is a comment
```

## Value Operations

### Value

Use to set or read a value:

```
Playback1.ClipContainer1.Opacity.Value = 0
```

### FadeValue

Fade to a new value in a defined time- first parameter is value, second parameter is time in seconds.

```
Playback1.ClipContainer1.Opacity.FadeValue 1,2
```

### ProgValue

Read or create a value in programmer mode:

```
Playback1.ClipContainer1.Opacity.ProgValue = 0
```

### FadeProgValue

Sets a new value in programmer and fades to this value - first parameter is value, second parameter is time in seconds.

```
Playback1.ClipContainer1.Opacity.FadeProgValue 1,2
```

## Assign Values

Use an equal sign for the following value operations:

### Value

```
Value = 1
Value = 2000
Value = 0.5
Value = 0,1,0.1 // <-- for e.g. DMX Values that consist of r,g,b
```



### Colors

```
//RGB
Value = #123 111 100
//ARGB
Value = #50 100 100 100
//RGB as Hex
Value = #FFFFFF
//ARGB as Hex
Value = #30FF00FF
//normalized RGB
Value = 0.5,0.5,0.5
//normalized ARGB
Value = 0.5,1,1,1
//normalized RGB
Value = {R:1.0,G:0.1,B:0.5}
//normalized ARGB
Value = {A:0.5,R:1.0,G:0.1,B:0.5}
Value = {R:10,G:100,B:50}
Value = {A:255,R:10,G:100,B:50}
```

### Checkboxes

*true and false*

```
Value = true
Value = false
```

### Settings

*Use the name that is used in inspector dropdowns and property fields.*

```
Value = FreeSync
Value = System2
```

### Allocate Text or URLs

```
Value = Hello World
Value = www.ioversal.com
```

## Return a Current Value

### **Value** (*without equals*)

*Entering Value (or ProgValue) returns the current value of an item.*

```
Playback1.Clipcontainer1.Opacity.Value
```

- the return value is e.g 0.5 when clip container's opacity currently is 0.5
- use return values also for an external request over the API

## Return A Value From An Item & Assign To Another Item

You can combine both variants - assign and return - to read out one item's value and assign it to another item.

Set notes from Clip Container 1 as Text into Text Content2

```
Content2.Settings.Text.Value = playback1.ClipContainer1.UserProperties
```

## Trigger Actions

- for example in a system:

```
System2.ControlViewer.Open
```

```
System1.WindowsShutdown
```

## Local Variables

When setting a local variable, there must be a single value (not a formula like the syntax for "Eval") on the right side of the equation.

```
Set cnt = 1  
Set cnt = Thisisatext
```

Assigning a value from a property to your local variable.

```
Set cnt = Playback1.ClipContainer1.Opacity.Value
```

### **Eval**

Defines or updates a local variable. The right side of the equation is evaluated as a formula (a so-called "expression").

```
Eval cnt = cnt + 1
```

Assign values of other variables.

```
Set max =100
Eval cont = cnt <= max
```

Enclose (nested) elements in square brackets (Unlike the syntax for Set).

Specify strings with single quotes.

```
Eval cnt = 'CNT: ' + Round([Playback1.ClipContainer1.Opacity.Value], 2)
// the variable cnt gets the (new) value based on a string and the rou
// opacity value from ClipContainer1
```

## Conditions

In the fashion of easy script commands, we have implemented simple conditional operations:

### **IfEqual**

Executes a specified script if parameter  $a = b$ . Parameters:  $a, b, \text{Script}$

```
IfEqual Sequence1.ClipContainer1.Opacity.Value, 0, Script2
```

### **IfExecute**

Executes a specified script if the specified expression is true.

```
IfExecute myval, Script1
```

### **IfGreater**

Executes a specified script if parameter  $a > b$ . Parameters:  $a, b, \text{Script}$

```
IfGreater Sequence1.ClipContainer1.Opacity.Value, 0, Script2
```

### **IfLesser**

Executes a specified script if parameter  $a < b$ . Parameters:  $a, b, \text{Script}$

```
IfLesser Sequence1.ClipContainer1.Opacity.Value, 1, Script2
```

### **IfUnequal**

Executes a specified script if parameter  $a$  is not  $b$ . Parameters:  $a, b, \text{Script}$

```
IfUnequal Sequence1.ClipContainer1.Opacity.Value, 0, Script2
```

## Tags

Defining and jumping to tags in a script:

### **:Tag**

Identifies and tags a line in a script. By using the Goto Tag command, script execution will be able to jump between tagged lines instead of going through the script lines top to bottom.

```
:start  
:part3  
:marker2
```

The tag name is preceded by a colon without a space. Tags must be placed at the beginning of the line that is supposed to be tagged. Examples can be found [here](#).

### **Goto Tag**

Jumps to the tagged line inside a script - in the actual script command, the tag name must be specified without a colon.

```
Goto part3
```

### **IfGoto** variable, tag

Continues to a specified tag if the specified expression is true.

```
Set Variable1= true  
IfGoto Variable1, start
```

## Special Commands

### **Wait**

Wait ## seconds

```
Wait 10
```

### **WaitAll**

Wait for all executing child scripts to finish.

```
WaitAll 10
```

### **Cancel**

Cancels all running scripts.

```
Cancel
```

### log

logs the returned value - the result is shown in [script monitor's](#) console and notification window  
use for e.g testing commands, preparing commands for the API...

```
log Playback1.Clipcontainer1.Opacity.Value
```

### log a text

```
log thisisatext
```

```
log this is a text
```

### logs a local variable or script parameter

```
Set variable1=50  
log variable1
```

## Call renamed items

An item can be called from a script by either its type-ID or its name. Both will be accepted.

### Example:

Device1 ( a 8 Bit Dimmer) was renamed to "Dimmer1"

The device can be called by both names:

```
Dimmer1.Settings.StartAddress.Value = 3  
Device1.Settings.StartAddress.Value = 3
```

## Indexers

In the context of Vertex scripts, indexers refer to a syntax that allows accessing several items at once, e.g.:

```
Playback[1-2].Play
```

Internally this is resolved to several script lines:

```
Playback1.Play  
Playback2.Play
```

This feature can be used as a powerful way to remote control multiple items with the touch of one button, for instance if indexers were used in a Control View script for a button/ slider etc..

Several Indexers can be used together, e.g.:

```
Sequence[1-2].ClipContainer[1-2].Opacity.Value = 1
```

This will be resolved to:

```
Sequence1.ClipContainer1.Opacity.Value = 1  
Sequence2.ClipContainer1.Opacity.Value = 1  
Sequence1.ClipContainer2.Opacity.Value = 1  
Sequence2.ClipContainer2.Opacity.Value = 1
```

Indexers can be used to access any scriptable objects. Even if a custom name is not specified, the objects are always accessible by their generic script name appended by their Script Id.

The indexer expression is enclosed in "[ ]" brackets and appended to the common or generic script name, as in the examples above.

An indexer expression can consist of several filter expressions, each separated by one or more spaces.

A filter expression can represent either a single value or a value range. To specify a range, concatenate the two values using "-" as seen in the examples above.



**values!**

**Important: do not add any spaces between the two range**

Valid values are either integers for filtering by Id or time codes for filtering Clip Containers by their playback position.

Value ranges must be of the same type, i.e., combining Id and time filters is not possible.

If the entire indexer expression or a value expression is a valid token expression (e.g. "Variable1.Value"), it will be evaluated before being further examined.

## Tokens that can be used in this context are:

- Any **properties** that return a value that can be evaluated here e.g.: 1, "1-2", "3:00:00".
- A **Playback** for filtering items using the current playback time.
- A **Sequence Cue** for filtering items using the cue's position.



**Note:** Local variables and parameters cannot be used, because the Indexers are evaluated before actually executing the script, using a common, global scope.

Each filter expression can optionally start with a prefix that acts as a „modifier“ and impacts how the expression is evaluated:

Modifier	Effect	Comments
-	<i>Exclude</i> value or range	Legacy syntax, identical to "!"
!	<i>Exclude</i> value or range	
+	<i>Include</i> value or range	Legacy syntax, obsolete
>	<i>Include</i> values greater than (or equal to)...	Not for ranges
<	<i>Include</i> values lesser than (or equal to)...	Not for ranges
*	<i>Include</i> selected items	

When using several filter expressions, an object is accepted if it matches **at least one inclusive** filter and **not any exclusive** filters.

Empty Indexer expressions will include all members (with the specified generic script name).

Invalid Indexer expressions will prevent the entire script from executing and throw an exception.

Valid Indexer expressions that yield no results during execution will be logged as a warning.

Notes regarding Clip Container filtering by time:

- Single time values will match if the time is within the Clip Container's range.
- "Greater than" filters will match if the Clip Container starts after the specified time.
- "Lesser than" filters will match if the Clip Container ends before the specified time.
- A specified time range will match if it overlaps with the Clip Container in any way.

## Examples

```
Sequence1.ClipContainer[].Opacity.Value=1
Sequence1.ClipContainer[1-5].Opacity.Value=0.2
Sequence1.ClipContainer[7 9].Opacity.Value=0.2
Sequence1.ClipContainer[<5].Opacity.Value=0.2
Sequence1.ClipContainer[1-2 9-10 4-7 -5-6].Opacity.Value=0.2
Sequence1.ClipContainer[5:0:0].Opacity.Value=0.2
Sequence1.ClipContainer[-Sequence1.Cue2].Opacity.Value=0.2
Sequence1.ClipContainer[Sequence1.Cue2].Opacity.Value=0.2
Sequence1.ClipContainer[>Playback1].Opacity.Value=0.2
```

*Setting new opacity value on selected items:*

```
Sequence1.ClipContainer[*].Opacity.Value=0.1
```

*Setting new opacity value on all clip containers on a specified track.*

```
Sequence1.ClipContainer[Sequence1.Tracks.Track2].Opacity.Value=0.1
```

*Changing size value of all click buttons on a specified ControlView page.*

```
ControlView1.Controls.ClickButton[ControlView1.Pages.Page1].Settings.S
```

*Changing size value of all click buttons on ControlView page1 and higher.*

```
ControlView1.Controls.ClickButton[>ControlView1.Pages.Page1].Settings.S
```

*Changing size value of all click buttons on ControlView page1 and page2. Pages listing separated by space.*

```
ControlView1.Controls.ClickButton[ControlView1.Pages.Page1 ControlView
```



## 6.12.2 Vertex Script Examples

Find below a list of the most common scripting commands available for VERTEX CORE objects as a reference.

### Jump to Categories:

- [Top/Root Level](#)
- [General Property Scripts](#)
- [System](#)
- [Systems Manager](#)
- [Backup Service](#)
- [Clip Container](#)
- [Programmer](#)
- [PME Playback](#)
- [Content](#)
- [Webbrowser Content](#)
- [Control View](#)
- [Devices](#)



### Show a list of all available script commands.

VERTEX comes with a dynamic script languages. Script commands are GUI-oriented in regards to the workflow.

**To show a list of all available script commands**, just click one of the script fields or into command section at the bottom and **press "CTRL and Space"**. A list of all available commands opens. Select an item from the list and press ENTER.

To show **all available commands for the next deeper level**, enter a decimal point "."

## TOP/ROOT LEVEL

**ProjectClose**

**ProjectLoad**

**ProjectSave**

**ProjectSaveBackup**  
**ProjectLoadPreviousBackup**  
**ProjectLoadPreviousAutoSave**

**ProjectSettings**

**ViosoCalibrate**  
**ViosoCalibrateAdv**  
**ViosoReCalibrate**  
**ViosoReApply**

**MainWindowMinimize**  
**MainWindowMaximize**  
**MainWindowExit**

*This will immediately exit & close VERTEX without saving.*

**Wait**

*Wait for a specified amount of time. Parameters: seconds (double)*

**Return**

*Stop script execution and return the specified result. Parameters: result*

**Log**

*Write to the script monitor's output window. Parameters: message*

**Clear**

*Clear the script monitor's output window.*

**WaitAll**

*Wait for all executing child scripts to finish. Parameters: timeout/seconds (double)*

**ListAllItems**

*Return a list of all scriptable (top-level) items.*

**Cancel**

*Cancel all running scripts.*

**ListItems**

*Returns a list of all items of a specific type.*

## Conditional Script Commands

### **IfEqual**

Executes the specified script if  $a = b$ . Parameters:  $a, b, \text{script}$

### **IfEqualEqual**

Executes the specified script if  $a = b$  and  $c = d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfEqualGreater**

Executes the specified script if  $a = b$  and  $c > d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfEqualLesser**

Executes the specified script if  $a = b$  and  $c < d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfEqualUnequal**

Executes the specified script if  $a = b$  and  $c \neq d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfExecute**

Executes a specified script if a local variable has a Boolean value of 1 or true

### **IfGoto**

[Jumps to a tag](#) if the condition - the value of the local variable - is true.

### **IfGreater**

Executes the specified script if  $a > b$ . Parameters:  $a, b, \text{script}$

### **IfGreaterGreater**

Executes the specified script if  $a > b$  and  $c > d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfGreaterLesser**

Executes the specified script if  $a > b$  and  $c < d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfLesser**

Executes the specified script if  $a < b$ . Parameters:  $a, b, \text{script}$

### **IfLesserLesser**

Executes the specified script if  $a < b$  and  $c < d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfUnequal**

Executes the specified script if  $a \neq b$ . Parameters:  $a, b, \text{script}$

### **IfUnequalGreater**

Executes the specified script if  $a \neq b$  and  $c > d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfUnequalLesser**

Executes the specified script if  $a \neq b$  and  $c < d$ . Parameters:  $a, b, c, d, \text{script}$

### **IfUnequalUnequal**

Executes the specified script if  $a \neq b$  and  $c \neq d$ . Parameters:  $a, b, c, d, \text{script}$

**PopupNotification**

Create a popup notification on top of the user interface. Usage: `PopupNotification WarningMessage`

**PopupNotificationWithClose**

Create a popup notification with close button on top of the user interface. Usage: `PopupNotificationWithClose WarningMessage`

## GENERAL PROPERTY Scripts

**ProgValue**

Get/set the property's value in the current programmer.

**FadeProgValue**

Fade the property's value in the current programmer. Parameters: `value (double), seconds (double)`

**SetProgRelative**

Add a relative value to the property's value in the current programmer.

**Value**

Get/set the property's value.

**FadeValue**

Fade the property's value. Parameters: `value (double), seconds (double)`

**SetRelative**

Add a relative value to the property's value.

## SYSTEM

**EnterFullScreen****LeaveFullScreen****FullScreenToFront****SmpeloSetMode****SmpeloSetTime**

*RunProcess*  
*RunProcessArgs*  
*KillProcess*  
*WindowsShutdown*  
*WindowsRestart*  
*ResetVideoInputs*  
*ForceResync*  
*ResetAsioDevice*  
*MouseMove*  
*MouseLeftClick*  
*StartVertex*  
*StopVertex*  
*ReconnectTimeServers*  
*RefreshNetworkAdapterList*  
*Connect*  
*Disconnect*

*NotchPurgeVRam*  
*RefreshDmxOutputs*  
*GetStatus*  
*LogStatus*  
*WriteToLog*

#### ***ScriptServerSendMessage***

*Send message as string. Byte or hex codes can be included like this {{72 101 108 108 111 32 0x57 0x6f 0x72 0x6c 0x64 0x13}}*

#### ***ScriptServerSendMessageToIp***

*Send message as string. Byte or hex codes can be included like this {{72 101 108 108 111 32 0x57 0x6f 0x72 0x6c 0x64 0x13}}*

## SYSTEMS MANAGER

*GetStatus*  
*AllSessionMembersEnterFullScreen*  
*AllSessionMembersLeaveFullScreen*

## BACKUP SERVICE

*Reconnect*

*Separate*

*TakeOverMasterRole*

## CLIP CONTAINER

*TempPreload*

*TempUnload*

## PROGRAMMER

*StoreAllData*

*ClearAllData*

## PME / PLAYBACK

**GotoCue**

*Transport playback to the specified cue. Parameters: cue number (int)*

**GotoCuePlay**

*Transport playback to specified cue and switches to Play. (Useful for Pause Cues)*

**LoadCue**

**ReleaseCue**

**GotoTime**

*Set Transport playback to the specified time.*

**GotoFrame**

*Transport playback to the specified frame.*

**IgnoreNextCue**

**EnableAllCues**

**GotoPrevCue**

**GotoNextCue**

*GotoFirstCue*  
*GotoLastCue*  
*GotoPreviousFrame*  
*GotoNextFrame*  
*Pause*  
*Play*  
*Stop*  
*TogglePlay*  
*GetPlaybackTime*  
*GetPlaybackCurrentCueTime*  
*GetPlaybackRemainingCueTime*  
*GetPlaybackTransport*  
*GetCurrentCue*  
*GetNextCue*

*FadeToCue*  
*FadeToCueHold*  
*FadeToCueHoldPlay*  
*FadeToCuePlay*  
*FadeToNexCue*  
*FadeToNexCuePlay*  
*FadeToPreviousCue*  
*FadeToPreviousCuePlay*  
*FadeToTime*  
*FadeToTimeHold*  
*FadeToTimeHoldPlay*  
*FadeToTimePlay*

## CONTENT

*SetVersion*  
*SetNextVersion*  
*SetPreviousVersion*  
*SetNextCycleVersion*  
*SetPreviousCycleVersion*

## WEBBROWSER CONTENT

### **NavigateTo**

*Navigate to new Url.*

### **Back**

*Navigate back.*

### **Forward**

*Navigate forward.*

### **Reload**

*Reload current Url.*

### **ReloadCache**

*Reload current Url and Cache.*

### **SendKey**

*Send Key: Left Right Up Down PageUp PageDown.*

### **SendJs**

*Send JavaScript Code to current page.*

## CONTROL VIEW

### **Close**

### **Open**

### **MoveBackward**

### **MoveForward**

### **GotoPage**

*Navigates to the specified page within the current control view. Parameters: page name (string), transition duration (seconds, optional)*

### **SetView**



Navigates to the specified view (and page). Parameters: view name (string), page name (string, optional), transition duration (seconds, optional)

**GetPage**

**GetView**

## DEVICES

### General Device Commands:

*Connect*

*Disconnect*

### StreamDeck:

*SetKeyColor*

### UDP Sender:

*ResetConnection*

*SendMessage*

### TCP Client:

*ResetConnection*

*SendMessage*

### TCP Connection:

*ResetConnection*

*Resets the connection.*

**SendMessage**

*Sends a string message to all connected endpoints. Byte or Hex Codes can be included like this {72 101 108 108 111 32 0x57 0x6f 0x72 0x6c 0x64 0x13} Use \{ or \} to send curly brackets.*

**SendMessageToIp**

*Sends a string message to a specific endpoints.*

**Process Device:****Start** *Start the process.*

*Stop* *Stop the process.*

**Maximize**

*Maximize the process main window.*

**Minimize**

*Minimize the process main window.*

**Restore**

*Restore the process main window.*

**PPT Device:****ResetConnection****CloseConnection****NextSlide****PreviousSlide****GotoSlide****SetVolume****Launch****Quit****StartPresentation****EndPresentation****OSC Sender:****ResetConnection**

*Resets the connection.*

**SendMessage**

*Sends an Osc message with one or multiple value. Usage: Address,Value,Value,...*

**SendBundle**

*Sends an Osc Bundle with one or multiple values. Usage: Address,Value,Value,...*

## KNX:

*ResetConnection*

*SendTrue*

*SendFalse*

*SendDpt9*

*SendDpt14*

## Kiosk Browser:

*Start*

*Stop*

*Maximize*

*Minimize*

*Restore,*

*NavigateTo*

*Back*

*Forward*

*Home*

*Reload*

*ReloadCache*

## Examples

*Start Playback1*

```
Playback1.Play
```

*Stop Playback 3*

```
Playback2.Stop
```

*Pause Playback3*

```
Playback3.Pause
```

Run Script 1

```
Script1
```

Show the notes of clip container 6 from sequence1 as text of text-content item "Text1"

```
Text1.Settings.Text.Value = Sequence1.ClipContainer6.UserProperties.No
```

Fade mix level of playback1 in PME live to full - fade time shall be 2 seconds

```
pme1.Playback1.MixLevel.FadeValue 1,5
```

Set opacity for clip container 1 to value 1

```
ClipContainer1.Opacity.Value = 1
```

Set background color of clip container 1 to color red =0.5, blue= 1, green = 1, alpha = 1

```
Sequence1.ClipContainer1.BackgroundColor.Value = 0.5,1,1,1 //normalize
```

Fade mix level of playback1 in PME live from to full - fade time should be 2 seconds

```
PME1.Playback1.MixLevel.FadeValue 1,5
```

Set network adapter for Art-Net™ on system 1 to "ETHERNET2"

```
System1.Settings.ArtNetAdapter.Value = ETHERNET2
```

Reset video inputs of system 2

```
System2.ResetVideoInputs
```

Reset video inputs of system 2.

```
System2.ResetVideoInputs
```

Change the label text of label 1 in control view 1.

```
ControlView1.Controls.Label1.Settings.Caption.Text.Value = "this is a
```

Perform a click on button 1 of control view 1.

```
ControlView1.Controls.ClickButton1.Click
```

Return the current page that is displayed by control viewer 1.

```
ControlViewer.GetPage
```

Switch to page 2 of the current control view. Also works as script for e.g. button in control view editor (run mode).

```
ControlViewer.GotoPage Page2
```

Delete label 1 on page 1 of control view1.

```
ControlView1.Pages.Page1.Label1.Delete
```

Assign content to a clip container:

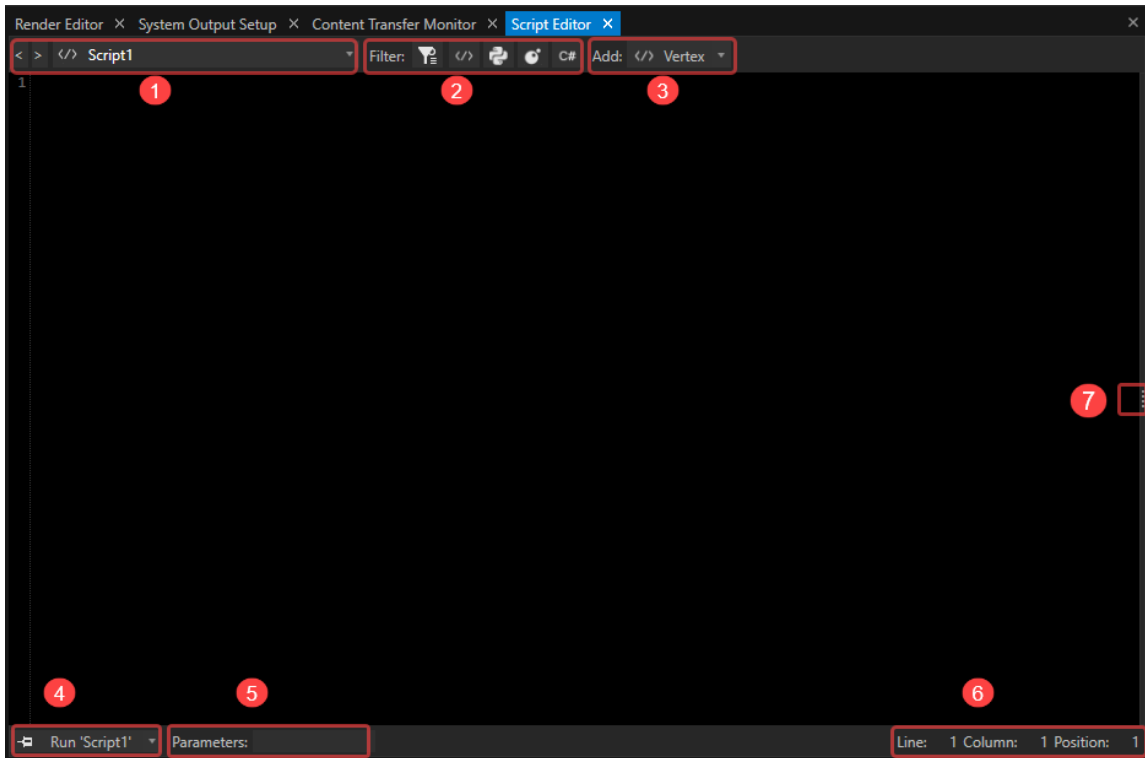
```
Playback1.ClipContainer1.MainContent.Value = Content1
```

## 6.12.3 Script Editor and Script Monitor

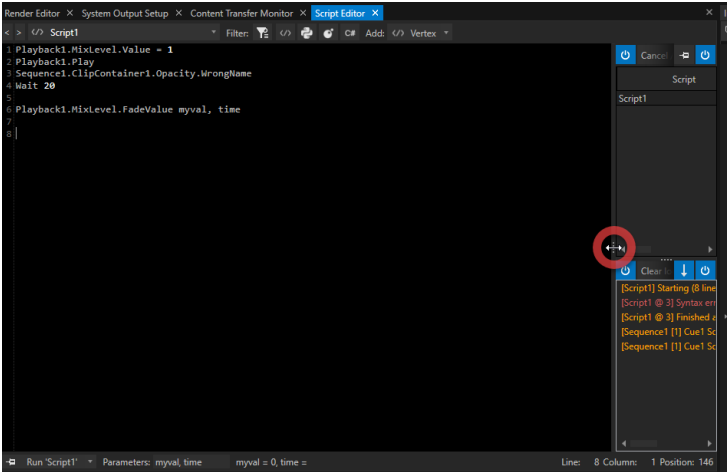
- VERTEX comes with **two powerful tools** that let you **write and manage scripts**.
- Create, write and run scripts in the **Script Editor**.
- Debug, monitor or log script commands and scripts with help of the **Script Monitor**.

### Script Editor

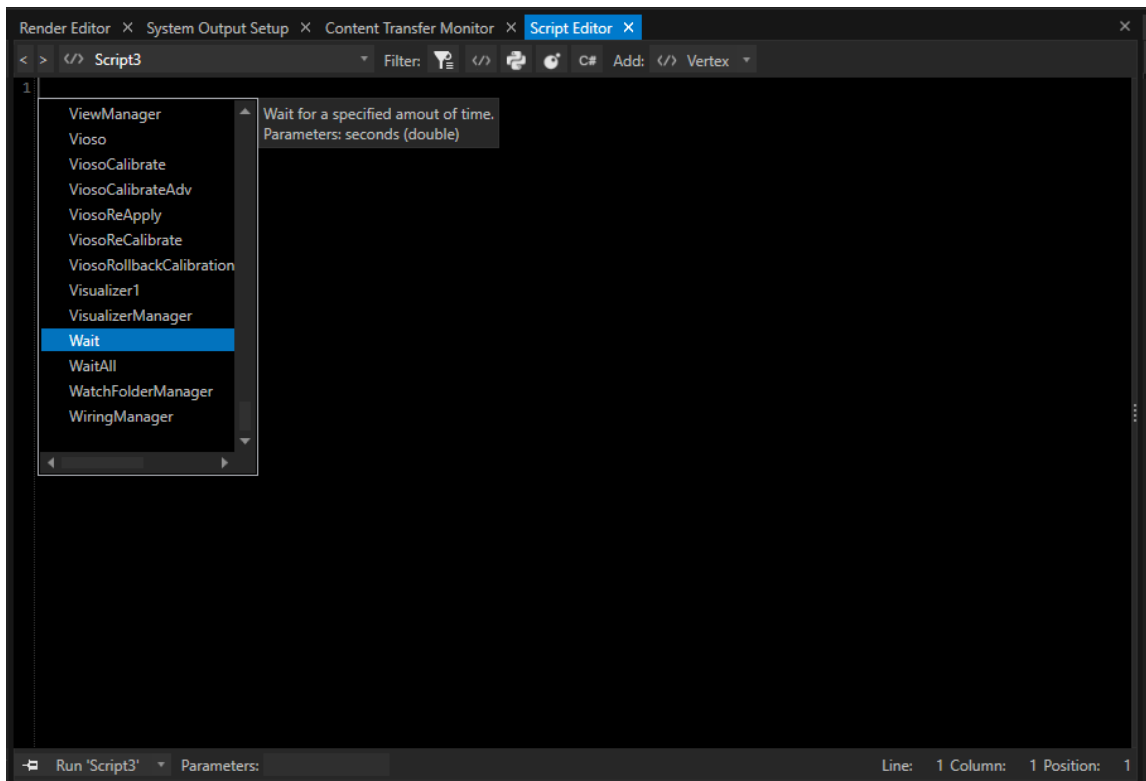
- This is where you write, test and edit your scripts.
- Let our [ScriptWizard](#) guide you through all available script commands.



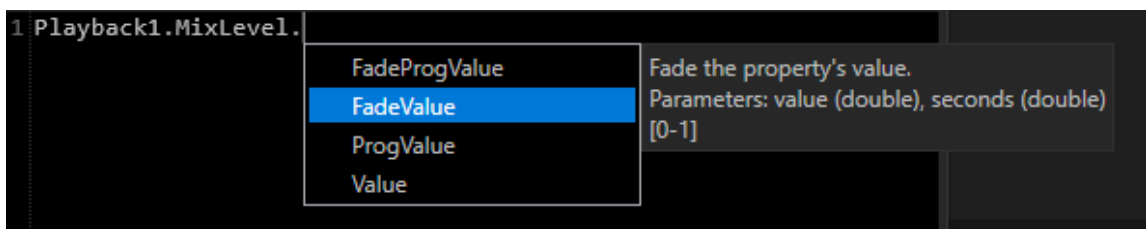
1	<b>History / Selection</b>	Select from a list of scripts or your history and display the script inside the editor.
2	<b>Filter Stand-Alone Scripts and Script Languages</b>	Disabled by default - script code from all elements in VERTEX is shown in the list. <a href="#">Read more here</a> . Enable to show only stand-alone <a href="#">scripts</a> .
3	<b>Create a Script</b>	Select your preferred <a href="#">scripting language</a> from the drop-down menu and create a new script.
4	<b>Pin Script for Running</b>	Pins the currently selected script to the Run command button right next to it.
	<b>Run Selected Script</b>	The currently selected script is started.
	<b><u>Run (with prompt)</u></b>	If there are defined <a href="#">parameters</a> for this script, you can assign values to them. A pop-up dialogue box will open to set values and optionally clear the cache.
5	<b><u>Parameters</u></b>	Enter Parameters for a Script. This is an easy way to set custom values that can be reused. <a href="#">Read more about parameters in the topic Scripts</a> .

		<p>This field is only available in stand-alone scripts. When you run a script with parameters, use the following syntax in the script command field:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <code>ScriptName Parameter1,Parameter2,Parame</code> </div>
<p><b>6</b></p>	<p><b>Cursor Orientation</b></p>	<p>This information on the whereabouts of your cursor will come in handy when writing long scripts.</p>
<p><b>7</b></p>	<p><b>Sidebar with Script Monitor</b></p>	<p>Pull out the <a href="#">Script Monitor</a> for a fast debugging and monitoring of your Scripts</p> 

## Working with the Script Editor

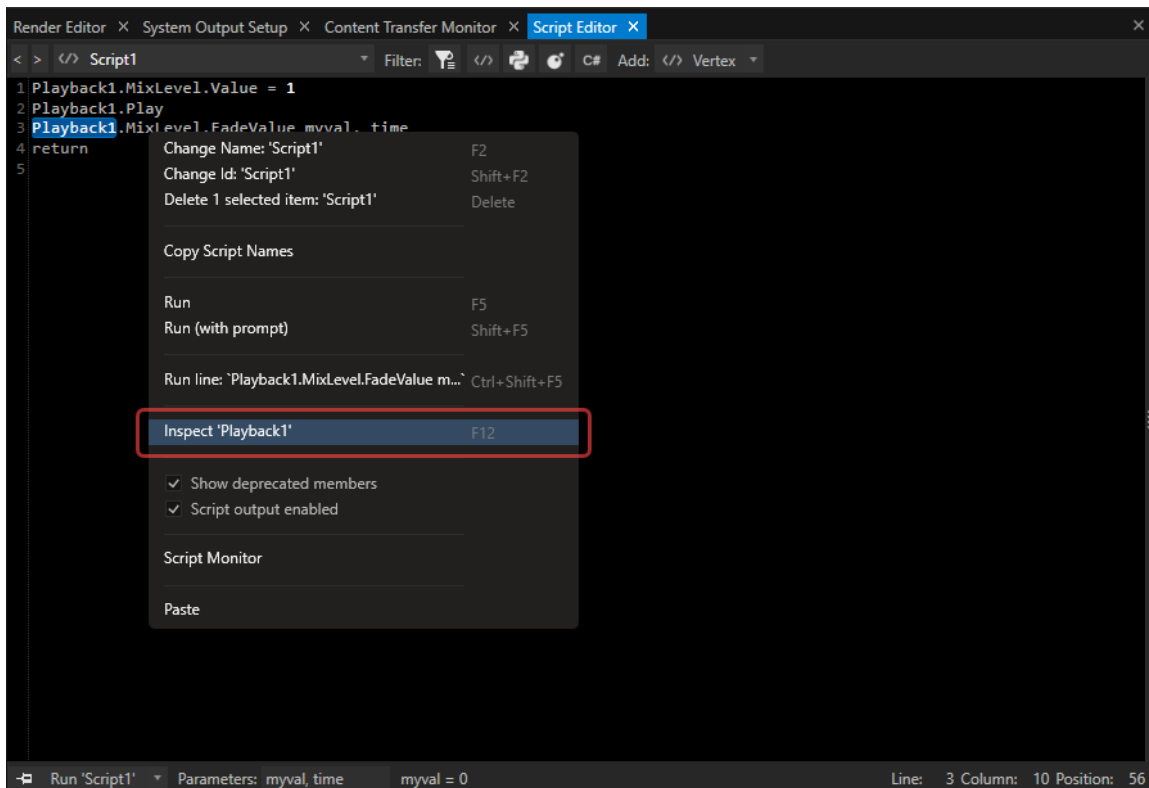


- The shortcut "**CTRL + SPACEBAR**" in an empty line opens a list of all available items and commands
- Navigate through the list with **UP/ DOWN ARROW KEYS** or **your mouse**. A tooltip will pop up showing expected parameters, syntax and range.
- Confirm your selection with **ENTER**.



- Enter a decimal point "." to open a list of all available child items and commands on the next deeper level.





- Right-Click with your Mouse on a line - you directly can **inspect the main element**.  
The selected Element is shown in the Inspector.



Run a Script with the help of Shortcuts

F5: Run

Shift+F5: Run (with prompt)

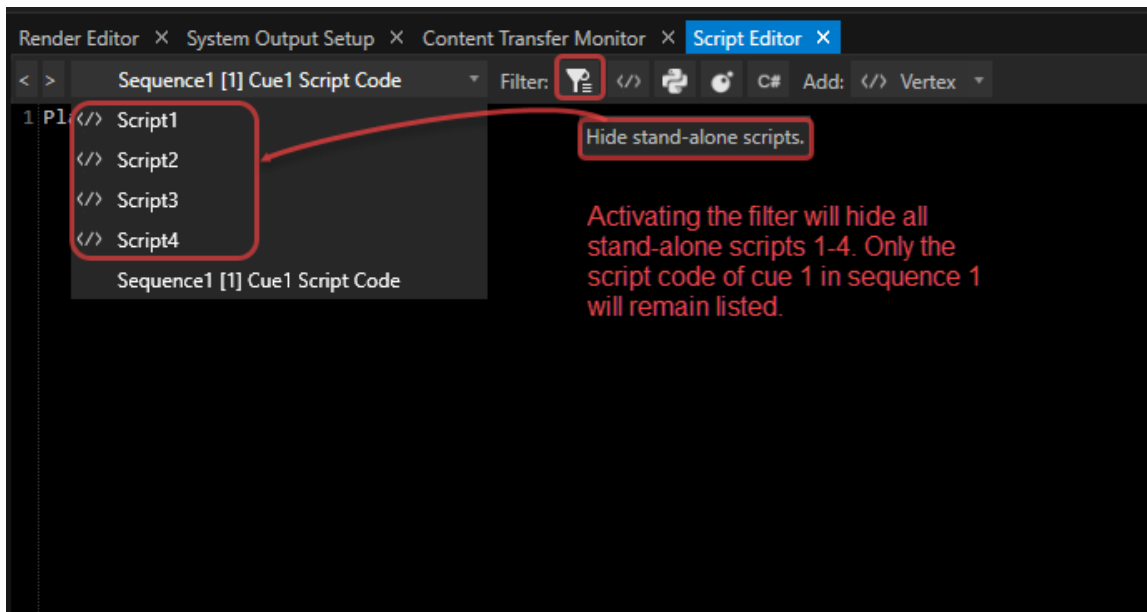
## Show Code From Other Script Command Fields in Script Editor

When the **filter symbol** in the **menu bar** is **deactivated**, the script editor shows **all scripts** that have been created in your project.

This includes the ones that are not stand-alone scripts but **connected to** other items in your project such as **clues, triggers, scheduled events and timers**.

The **dropdown list** of the script editor is now extending to all other available scripts in your project.

The **name of such a script** is **composed of the main and child element** - for example: "Sequence1 Cue1 Script Code".



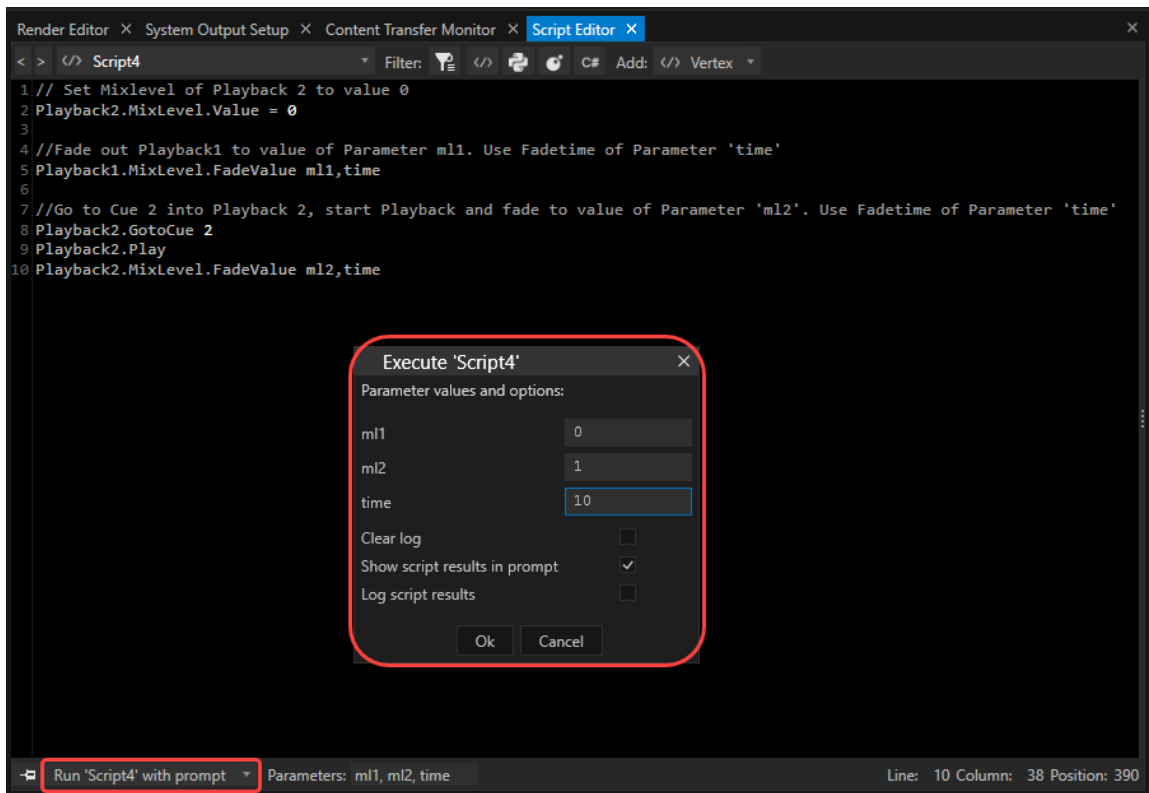
Likewise, you can show/ hide scripts of all available scripting languages by activating their respective filter symbols:



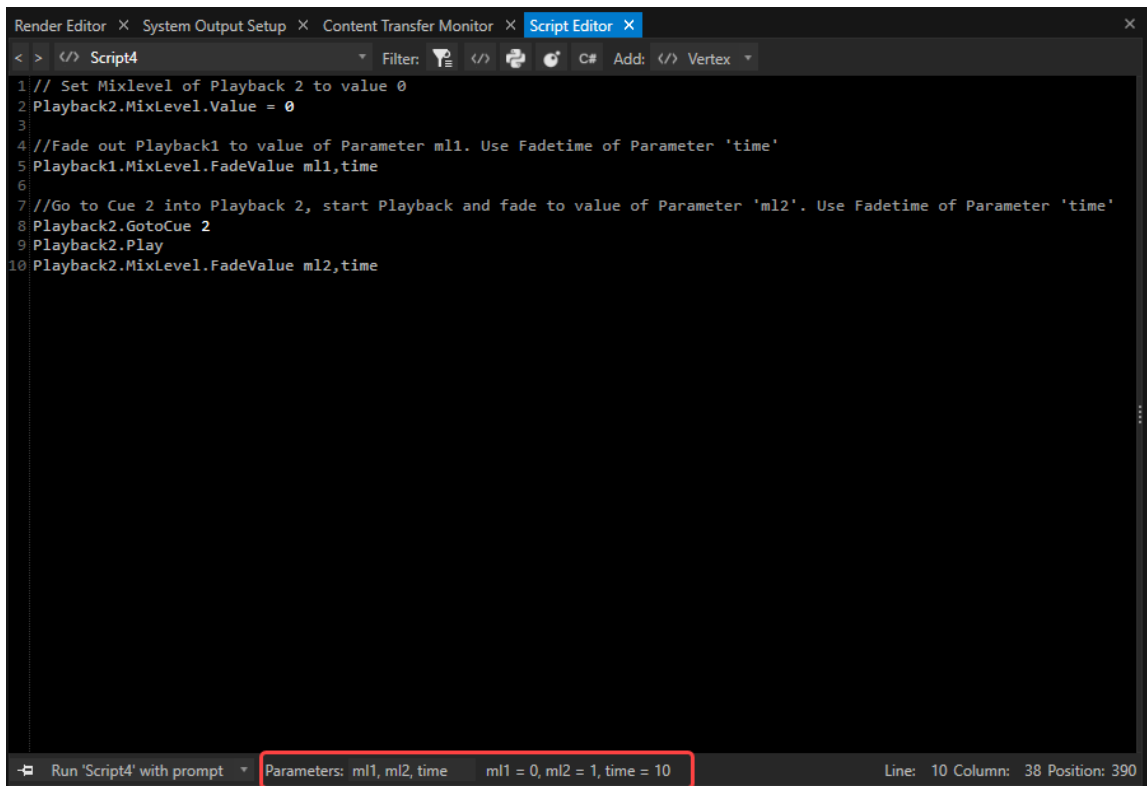
## Run Scripts with Prompt

For testing purposes, you can add parameters directly from inside the script editor by using the button: Run 'Script' with prompt.

A popup window opens, where you can add values for all expected parameters.



Once parameters are assigned via the prompting window, their values are listed next to the Parameters field:



The screenshot shows a software interface with a 'Script Editor' window. The window title bar includes 'Render Editor', 'System Output Setup', 'Content Transfer Monitor', and 'Script Editor'. The script content is as follows:

```
1 // Set Mixlevel of Playback 2 to value 0
2 Playback2.MixLevel.Value = 0
3
4 //Fade out Playback1 to value of Parameter m1. Use Fadetime of Parameter 'time'
5 Playback1.MixLevel.FadeValue m1,time
6
7 //Go to Cue 2 into Playback 2, start Playback and fade to value of Parameter 'm2'. Use Fadetime of Parameter 'time'
8 Playback2.GotoCue 2
9 Playback2.Play
10 Playback2.MixLevel.FadeValue m2,time
```

At the bottom of the window, there is a status bar with a dropdown menu set to 'Run 'Script4' with prompt'. To its right, the parameters are displayed as 'Parameters: m1, m2, time' and 'm1 = 0, m2 = 1, time = 10'. The status bar also shows 'Line: 10 Column: 38 Position: 390'.

## Script Monitor

- *all scripts that are currently running* can be monitored
- *cancel, pause or resume* any running script
- a **console** window lists events, status and errors
- a *log script* command passes values as lines into the **script monitor console**.

## Open the Script Monitor

There are two ways to open a Script Monitor:

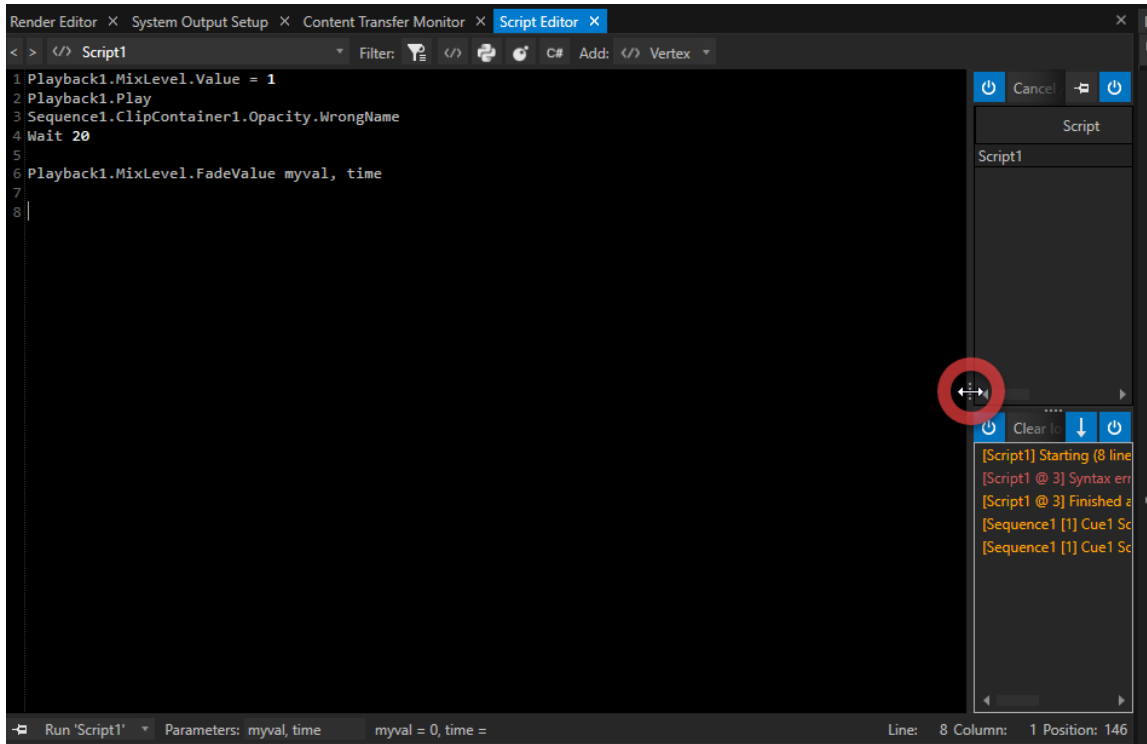
### 1. As an additional window

- go to the "Windows" Tab of the main menu on top
- open a new Script Monitor Window

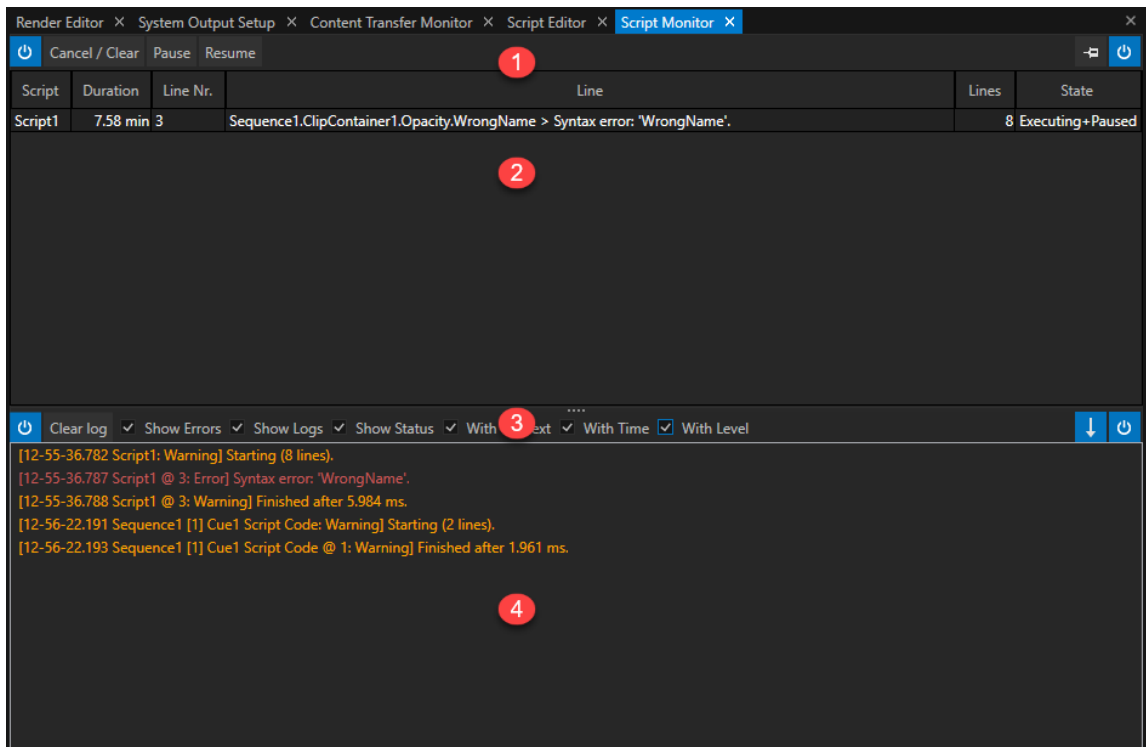
### 2. Pull out as Sidebar in Script Editor

- move the mouse to the right edge of the script editor

- touch the symbol with the four dots there
- hold the mouse and drag it to the center of the editor



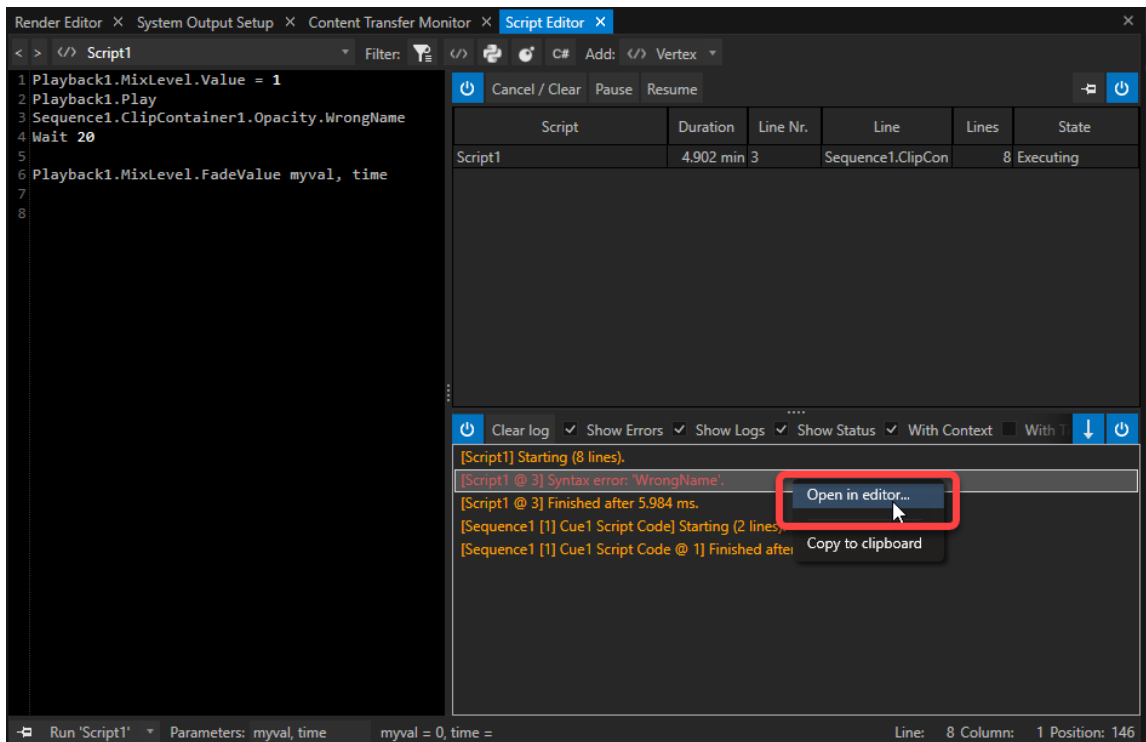
## User Interface



<p><b>1</b></p>	<p><b>Script Monitor Control</b></p>	<p><b>Power Button</b> enables/disables script monitor  <b>Cancel</b> script processing /clear monitor  <b>Pause</b> a selected script  <b>Resume</b> a selected script</p>
<p><b>2</b></p>	<p><b>Active Scripts</b></p>	<p>- lists all active scripts that are running, paused or stopped at an error          - finished scripts are no longer displayed in the list</p>
<p><b>3</b></p>	<p><b>Console Control</b></p>	<p><b>Power Button</b> - enables the script monitor console          - disable to save processing power  <b>Clear</b> the console and restart with an empty console  <b>Filter</b> information to be logged from the console:  <b>Errors</b> (in red)  <b>Logs</b> ( e.g. Script 4@5)  <b>Status</b> (yellow script lines with only information)  <b>With Context</b> (what Script and line)  <b>With Time</b> (show/ hide timestamps)  <b>With level</b> (e.g. warning)</p>
<p><b>4</b></p>	<p><b>Script Console</b></p>	<p>area where script execution info is logged</p>

*ScriptMonitor* as a sidebar in **ScriptEditor** enables you to **directly jump to the relevant line** in a script:

1. **rightclick on a script monitor or a console entry**
2. select **"open in editor"**

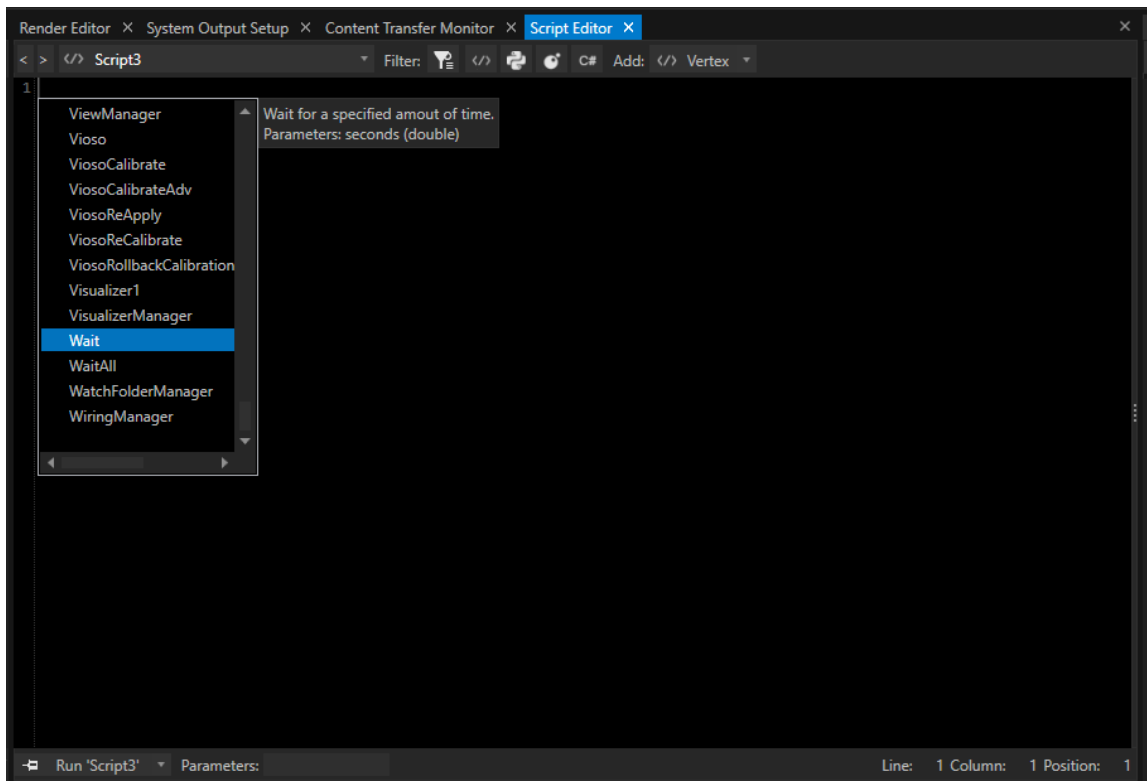


## 6.12.4 Scripts

- Combine **any number of script commands into one script**.
- Write **complex tasks** and trigger them by script with only one command.
- **Define parameters** and run a script with their respective values.

## How To Create Scripts In VERTEX

- create a new script via the main menu **CREATE** tab
- or
- open a new **Script Editor** from main menu **WINDOWS** tab
- Create a new script by clicking on the **"Add"** button



You are allowed to **rename a script from its default name** to a custom name such as `PlayAll`.

If renamed, you can use either `Script1` or `PlayAll` to call and run the script.



#### Script Names

Script names must contain at least one letter. Allowed are letters [a-z], numbers [0-9] and `_`.  
Not allowed are mathematical operators like `*`, `+`, `,` `/`

Press **Shortcut CTRL+ Space** at the start of an empty line to open the ScriptWizard: a list of all available commands and elements will be displayed.

**Enter a dot** after to show a list of all available properties or command for this item.

For **detailed syntax and structure**, please read the topic [Vertex Scripting](#).

## Examples



Below you will find examples to help you getting started with scripting.

The prefix // separates comments from script commands.

```
//Assign a new URL to HTML Content 3
Content3.Settings.Url.Value = www.ioversal.com

//Wait 3 Seconds
Wait 3

// Go to Cue 2 in Playback1 and start playback
Playback1.GotoCuePlay 2
```

```
// Set Mixlevel of Playback 2 to 0
Playback2.MixLevel.Value = 0

//Fade out Playback1 with fade time 3 seconds
Playback1.MixLevel.FadeValue 0,3

//Go to Cue 2 into Playback 2, start Playback and fade in
Playback2.GotoCue 2
Playback2.Play
Playback2.MixLevel.FadeValue 1,3
```

```
//Backup Script Example for a Session Member

// Set System2 as Render and Audio Backup for System1
System1.Settings.RenderSystem.Value = System2
System1.Settings.AudioSystem.Value = System2

//wait 3 seconds
wait 3

// Lightware Matrix as Device from Library - switch Input 4 to Output
Device1.Connect
Device1.RouteInputToOutput 4,2
```

```
//Working with tags inside a script and jump to tags inside a script

Playback1.Play
wait 2
//jump to tag start
Goto Start

//tag position3
```

```
:position3
wait 1
Playback1.GotoCue 3
wait 2
//jump to tag end
Goto End

//tag start
:Start
Playback1.GotoCue 2
wait 3
Goto position3

//tag end
:End
Playback1.Stop
```

The [Script Monitor](#) helps you with debugging your scripts and the search for errors.

## Run a Script

There are **different options** how to run scripts in VERTEX suiting your project and workflow:

1. Run a Script directly out of the **script editor** by pressing the "**Run**" button
2. Use the **script's name** in **each scripting field in VERTEX**
3. Type the script's name into the **command line** and confirm with ENTER.



### **Running scripts that have been renamed**

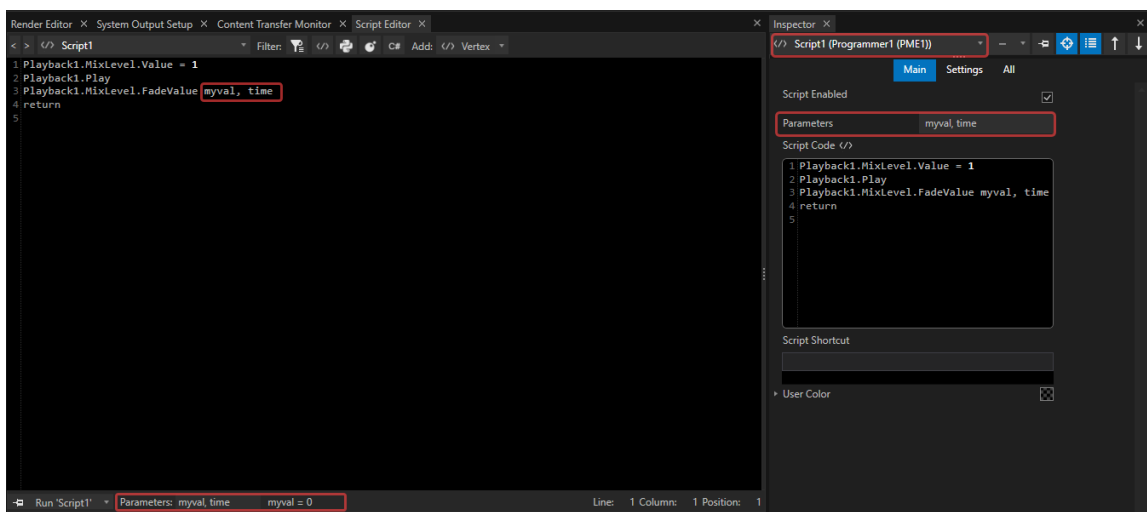
If a script has been renamed, use either the default name of the script or its new name.

For example: **Script1** was renamed to **PlayAll**. To run it, type into a scripting field or the command line either **Script1** or **PlayAll**.

## Parameters

- If you **define one or multiple parameters for a script**, you have the option to **turn over values when running a script**.
- Parameters can be used **with their name as local Variables inside a Script**.

## Define Parameter Names



- Use the **Parameters field in the script editor** to define the parameters for the selected script.  
or
- Select a script **in the inspector** and enter the parameters there.



### Define and use

Multiple Parameter names must be separated by a comma followed by a space character.  
Parameter names can be used in the same way like local variables inside a Script

### Example for Parameters and their use as local Variable inside a Script:

Script 1 has the parameters "m1", "m2" and "time" assigned

```
// Set Mixlevel of Playback 2 to value 0
Playback2.MixLevel.Value = 0

//Fade out Playback1 to value of Parameter m1. Use Fadetime of Parameter m2
Playback1.MixLevel.FadeValue m1,time

//Go to Cue 2 into Playback 2, start Playback and fade to value of Parameter m2
Playback2.GotoCue 2
Playback2.Play
Playback2.MixLevel.FadeValue m2,time
```

## Run a Script with Parameter Values

When you run a script with parameters, use the following syntax in every script command field:

```
ScriptName Parameter1,Parameter2,Parameter3
```

For example the script above with 3 parameters "ml1", "ml2" and "time":

```
Script1 0,1,5
//Assigns the Values 0 for parameter 'ml1', 1 for 'ml2' and 5 for para
```



**For testing purposes, you can add parameters directly from inside the script editor by using the button: Run 'Script' with prompt .**

A popup window opens, where you can add values for all expected parameters.

## 6.12.5 Other Scripting Languages

Here are a few examples on how to work with other scripting languages in VERTEX:

- [Lua Scripting](#)
- [Python Scripting](#)
- [C# Scripting](#)

### 6.12.5.1 Lua Scripting

Sample Code

```
-Vertex Lua Sample Code
- Pass input123 parameter to variable
var1 = input1

- Run Vertex Script
Script.Run ('Log Hello Lua')

- Get a return value from Vertex Scripting
var2 = Script.Run('Return SystemsManager.GetStatus')

-[[Return a value to Vertex Scripting]]
```

```
return var1
```

*Script.Run lets you interact from Lua with VERTEX scripting.*

### 6.12.5.2 Python Scripting

Sample Code

```
#Vertex Python Sample Code
#Pass input parameter to variable
myVar = input1

#Call a Vertex Script from Python script code
Script.Run('Log Hello Python')

myVar2 = 'System Status'

#Get a return value from Vertex Scripting
myVar3 = Script.Run('SystemsManager.GetStatus')

#Use returnValue to return a value to Vertex Scripting
myVar4 = 12
returnValue = myVar * 2
```

*Script.Run lets you interact from Python with VERTEX scripting.*

### 6.12.5.3 C# Scripting

Sample Code

```
//Vertex C# Sample Code
//Pass input parameter to variable
var input = input1;

for(int i = 0; i < 10; i++)
{
    //Call a Vertex Script from C# script code
    Script.Run("Log Hello C# " + i);
    System.Threading.Thread.Sleep(500);
}
```

```
//Get a return value from Vertex Scripting
var r = Script.Run("Return SystemsManager.GetStatus");

//Return a value to Vertex Scripting
return "Return from C#";
```

C# Engine interacts with VERTEX Script commands.

`Script.Run` lets you interact from C# with VERTEX scripting.

## Compiling Script.Run Calls

`Script.Run` allows the passing of multi line scripts.

### previous code:

```
Script.Run("Playback1.Active.Value = true");
Script.Run("Playback2.Active.Value = true");
Script.Run("Playback3.Active.Value = false");
Script.Run("Playback4.Active.Value = false");
```

### new code:

```
Script.Run(@"Playback1.Active.Value = true
           Playback2.Active.Value = true
           Playback3.Active.Value = false
           Playback4.Active.Value = false");
```

The "@" (verbatim string literal) is important for adding multiline inside String-Literals. Alternatively you can combine them instead of the "@" with "\n".

## Script.Run Calls With Parameters

`Script.Run` has now got its signature changed allowing for optional parameters:

```
public static object Run(string script, params object[] parameters)
```

### previous code:

```
Script.Run("Playback1.FadetoCuePlay " + newMessage.ToString());
```

```
new code:
    Script.Run("Playback1.FadetoCuePlay", newMessage);
```

Currently, this just saves the time to concatenate the strings manually. In the future we'll be able to optimize these calls efficiently.

However, this does not work with multiline scripts.

## void Log(text)

The use of `Script.Log` command is simpler and more efficient than before:

```
previous code:
    Script.Run("Log Status: "+message);

new code:
    Script.Log("Status: "+message);
```

Omit the "Log" before "Status".

## void Script.SetVariable(name, value)

```
previous code:
    Script.Run("Var_MyVal.value = My Value");

new code:
    Script.SetVariable("Var_MyVal", "My Value");
```

## object Script.GetVariable(name)

*T* `GetVariable<T>(string name, T defaultResult = default)`

```
previous code:
    String MyValreturn = Script.Run("return Var_MyReturnVal.Value").To
```

```
new code:
    var MyValreturn = Script.GetVariable<string>("Var_MyReturnVal");
```

The generic variant takes over the necessary value conversion (by internal converters).

## bool Script.TryGetVariable<T>(name, out T value)

This is another function helping to avoid extensive concatenations of Script.Run strings as well as avoiding IfExecute from within C#.

It is just way more efficient using C#'s native logic:

```
previous code:
    Script.Run("IfExecute Var_MyVal.Value, Device1.SendMessage "+message);

new code:
    if (Script.GetVariable<bool>("Var_MyVal")) Script.Run("Device1.SendMessage "+message);
```

## 6.12.6 Variable

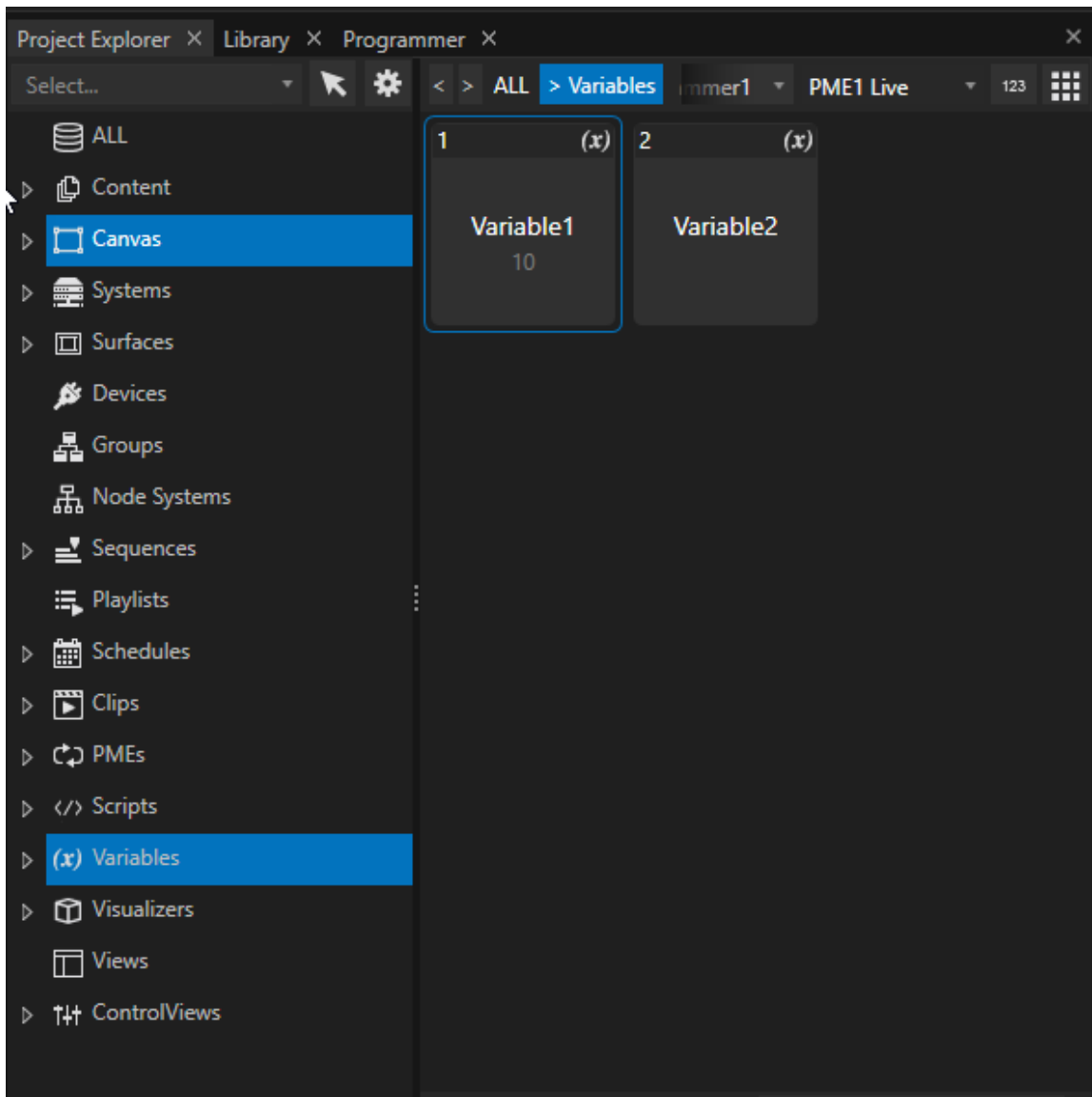
- Use a variable for **recurring tasks** and assign a value to it.
- **Global variables** are listed in the **project explorer**.
- **Local variables** are only used in a **single script**.
- Use variables **in script commands** instead of fixed numbers.

## Global Variables

### Creating a Variable:

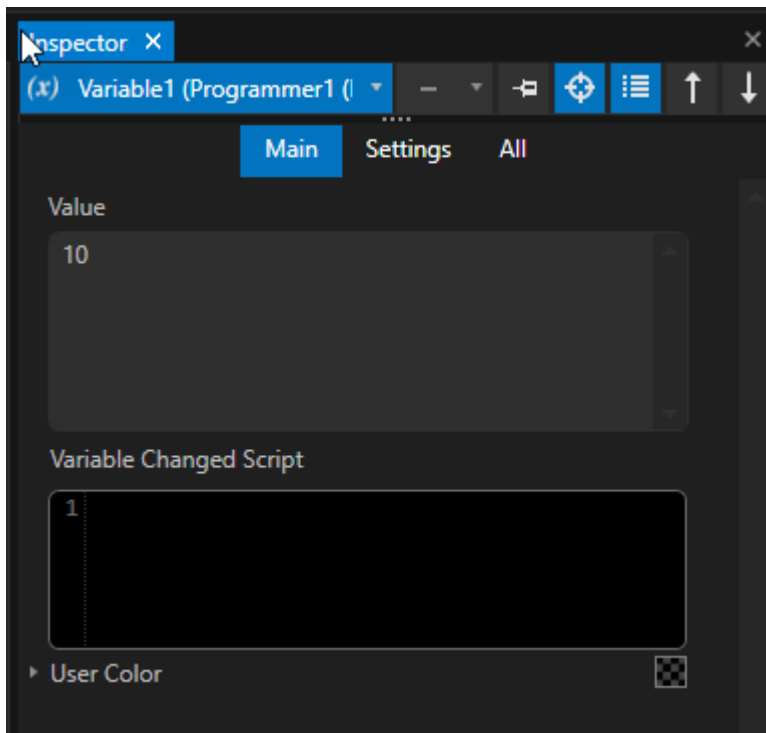
- Go to MAIN MENU -> CREATE -> Variable
- A new section is created in the project explorer called Variables.
- This is where the newly created variable is listed.





## Assign a Value

*The current value of a variable can be assigned in the inspector:*



Of course, you can assign **values** to a variable **via script commands** in any **scripting field** or in the **command section** of the status bar:

`VariableName.Value = ##`

```
Variable1.Value = 10
```

## Use a Variable

Use a variable in script commands instead of the expected value:

Without Variable

```
Playback2.MixLevel.FadeValue 1, 10
```

With Variable

```
Playback2.MixLevel.FadeValue 1, Variable1.Value
```

You are allowed to **rename a variable from default name** to let's say `FadeInTime`.

If renamed, you can use either `Variable1.Value` or `FadeInTime.Value` in script commands.

## Local Variables

- *Local Variables are only valid in a single script.*
- *You can assign a value to your local variable and use it on different lines in a script.*

### Create a Variable in a Script Command

- *All the information about **syntax** can be found here: [Script Command - Local Variables](#)*
- *Local variables **are only valid in their particular script**. Unlike global variables, it is impossible to use them in other scripts or with other script commands.*
- *The variable's name is only valid for this particular script.*

### Script Parameters

- *It is possible assign parameters to a script.*
- *These parameters can be used as local variables in the script.*
- *Read more about parameters for scripts: [Run a Script with Parameters](#).*

## 6.13 Timer

VERTEX's timer function allows the display of running and remaining time, as well as triggering 4 separate scripts by countdown:

- at the start of a timer
- once the timer is finished
- if the timer is set to restart the countdown in a loop
- for when the timer is stopped

An example for a possible application would be a museum setup, where visitors pass a triggering device that starts the timer prompting a cue or playback. Or a Clip Container's end script can trigger the start of a timer to tell the audience the time remaining until the next show during an intermission.

## How It Works

The screenshot illustrates the VERTEX STUDIO SUITE interface for configuring and running a timer. Key components and annotations include:

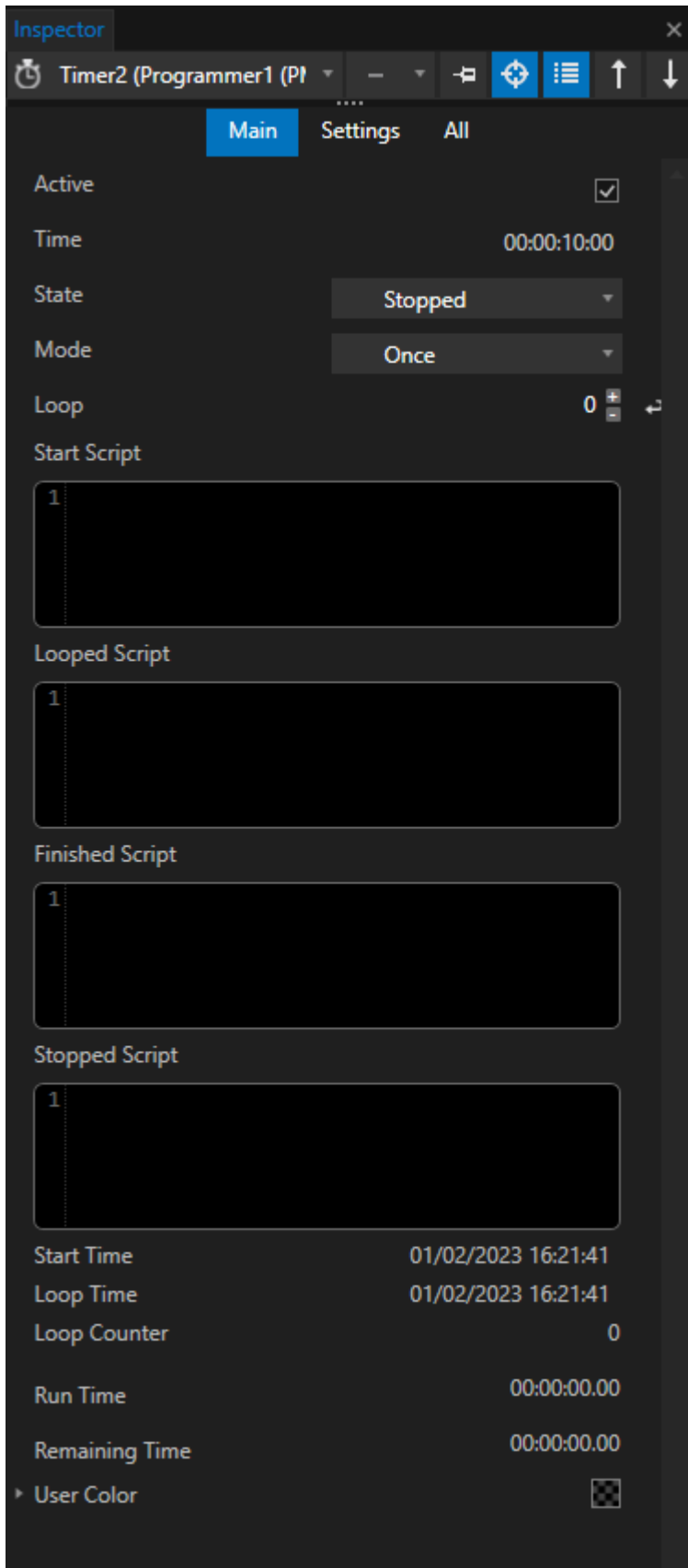
- Main Menu:** A red box labeled "create a timer from Main Menu" points to the "Timer" icon in the main menu.
- Canvas:** The central canvas displays "Timer STARTED 00:00:05.31".
- Playback Editor:** The bottom editor shows a timeline with a red box labeled "try starting the timer from command line" pointing to the "Cmd: Timer1 Start" command.
- Script List:** A red box labeled "this timer's scripts change the text content for example" points to the script list in the settings panel.
- Settings Panel:** A red box labeled "incorporate the source property into the text content with {0}" points to the "Source Property" field, which contains "{0} Remaining Time". Another red box labeled "this timer's values are being passed into the text content" points to the "Remaining Time" field in the settings.

The example above explained:

- A **timer** is being **created** by going to MAIN MENU > CREATE > TIMER.
- Pin the timer to a new Inspector via the context menu (right click on the timer tile in Project Explorer).
- Create a **text content** (either MAIN MENU > CREATE > CONTENT or context menu on Content in Project Explorer).
- Go to the **timer's Inspector**, drag Run Time or Remaining Time and drop it into the **text content's Inspector** Source Property field.
- Drag & Drop the text content into the Sequence and the timer will now be displayed as text content. If you wish to write additional text, it will override the source property display. Incorporate the value back into the text body by adding "{0}".
- Create more text content to display all timer states from Started (running) to Stopped.
- Each timer-script in this example changes the text content of ClipContainer1 so that the displayed text will reflect the changed timer status (started, finished, etc.):  
Start Script: `Playback1.ClipContainer1.MainContent.Value = Content1`  
- text content says *TIMER STARTED* plus time display.  
Finished Script: `Playback1.ClipContainer1.MainContent.Value = Content2`  
- different text content says *TIMER FINISHED* plus time display.
- Start the timer with the command line: `Timer1.Start` or use this script command elsewhere- i.e. in a Control View Editor on a button.

## Settings

In the **timer's Inspector**, the following settings are available:



<b>Active checkbox</b>	enables / disables the timer.
<b>Time</b>	sets the duration of the countdown with a double-click on the digits.
<b>State</b>	drop-down menu automatically updates the current status of the timer and allows for manual changes as well. Possible timer states are: <b>Running</b> , <b>Paused</b> , <b>Finished</b> and <b>Stopped</b> . Changes in a timer's status will execute the timer's scripts.
<b>Mode</b>	has got two possible settings: <b>Once</b> (default) and <b>Loop</b> . If set to <b>Loop</b> , the timer will start again after it has finished counting down. <b>Loop</b> sets the number of times the cycle restarts.
<b>Start Script</b>	is executed at the start of the timer.

<b>Looped Script</b>	<i>is executed at the beginning of each loop cycle.</i>
<b>Finished Script</b>	<i>is executed once the countdown is finished.</i>
<b>Stopped</b>	<i>is executed when the timer is stopped.</i>
<b>Start Time</b>	<i>captures date and time when the timer was last started.</i>
<b>Loop Time</b>	<i>captures date and time when the last loop cycle was started.</i>
<b>Loop Counter</b>	<i>counts the number of loop cycles executed.</i>
<b>Run Time / Remaining Time</b>	<i>captures the current running /remaining time which <b>can be exported as a value</b> in scripts or be simply displayed in text content.</i>
<b>User Color</b>	<i>customizes the timer tile's color.</i>





## 6.14 SMPTE

- ioversal's **SMPTE Timecode LTC [USB Interface](#)** is designed to receive or transmit timecode and is available through our website.
- **[Synchronize your playback](#)** to incoming timecode.
- **[Generating timecode](#)** with **SMPTE clip containers** out of your sequence has never been easier.
- The acronym SMPTE stands for Society of Motion Picture and Television Engineers and is a common synonym for timecode.

### LTC from "Audio" .WAV Files

Beware, if you cannot avoid some edge cases that may require working with timecode that comes from a .WAV file. While it is not impossible synching to LTC that is coming from an audio file and output, caution is well advised regarding the following:

- Sample rate & format needs to match from source to target. Meaning, if the source system's audio interface is running with 48kHz/ 16bit, the target system should do the same.
- Moreover, the .WAV file that contains LTC should be created in 48kHz/16bit as well.
- When routing LTC to your Windows audio outputs, go to Windows Sound Settings > Device Properties > Additional Device Properties and make sure to disable the Signal Enhancements of your output device.

Settings

### Device properties

Windows Audio Rename

Disable

#### Spatial sound

Spatial sound format

Select your spatial sound format for an immersive audio experience that simulates a more realistic environment.

Off

Balance

L 100

R 100

#### Related Settings

[Additional device properties](#)

[Get help](#)

[Give feedback](#)

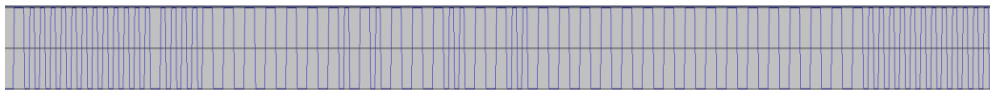
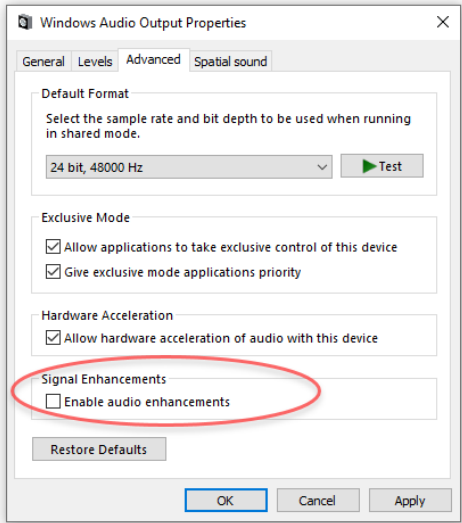


Fig. 1 shows what the waveform looks like generated by ioversal's SMPTE Timecode LTC USB Interface.

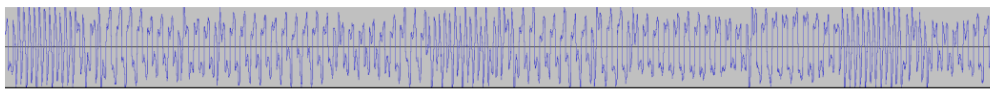


Fig. 2 shows a waveform from sampled audio input originating from timecode in a .WAV file.

*Timecode Generator Waveform vs .WAV File Timecode*

Bottom line: **the best results in frame-tight accuracy is achieved with a special timecode interface or generator.** Beware of the possible issues when working timecode via audio routes.

## 6.14.1 SMPTE IO Interface Configuration

ioversal offers a driverless plug and play USB SMPTE interface to input or output SMPTE timecode:



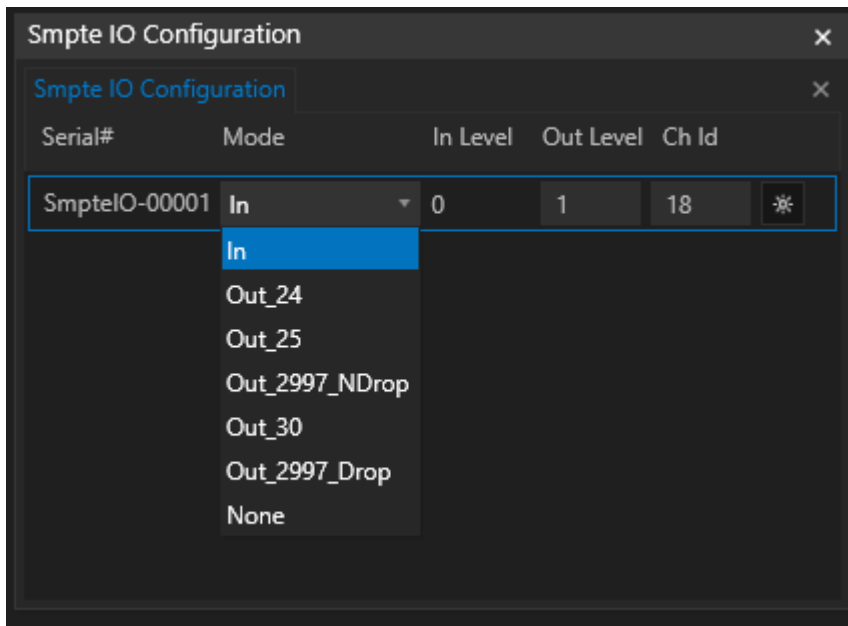
### **Purchase a SMPTE LTC USB Interface**

You can buy a SMPTE IO on [www.ioversal.com](http://www.ioversal.com) - or just contact an [ioversal partner](#)

- *ioversal's SMPTE interface needs to be configured before first use.*
- *Channel ID Settings are stored on the USB-Interface. Once configured for a project, the interface can be plugged and played.*

1. *go to Main Menu -> Windows -> SMPTE IO Configuration*
2. *connect the SMPTE IO interface to one of your local Systems USB ports, this interface should now be listed*
3. *select whether you want to send or receive timecode*
4. **set an ID for the interface** - *you will need this ID later to specify which interface sends or receives your timecode.*

*When working with more than one SMPTE IO Interfaces, each interface needs a unique ID.*



## 6.14.2 Receiving Timecode

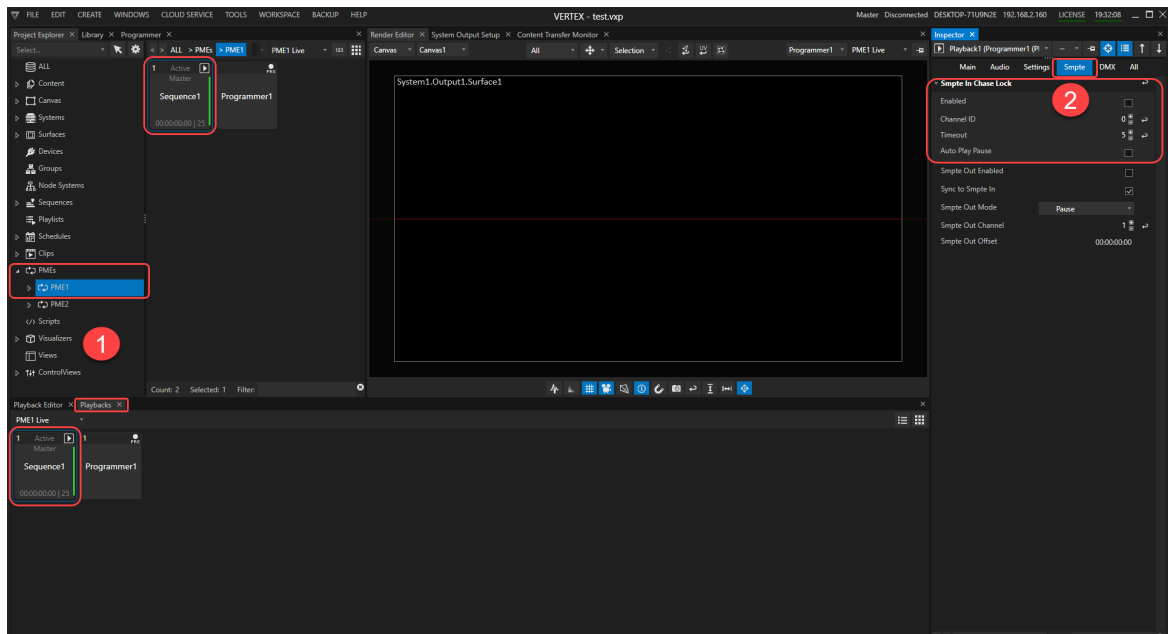
### Synchronize VERTEX to external Timecode with *SMPTe In Chase Lock*

- This feature enables VERTEX to continuously chase external SMPTE LTC and lock to it.
- No need to synchronize your system clock to external SMPTE IO - your system can still sync to your favorite clock while the playback is syncing to external LTC.
- Super efficient and fast when running in session mode - session members will be able to resynchronize to LTC changes in <1sec.



**Important notice when running in session mode:**

the **SMPTe IO USB interface** receiving external timecode **needs to be connected to** (and **configured** for) **the master system.**



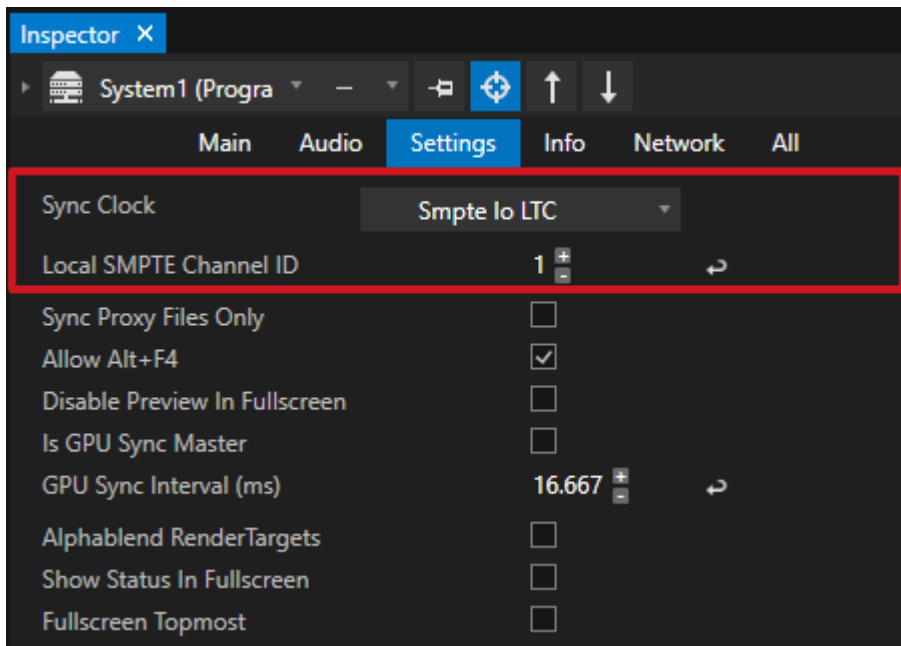
Let's walk you through accessing the settings for the SMPTE In Chase Lock function in 3 little steps:

- 1) select your active playback either from Project Explorer -> PMEs or Playback tab next to the playback/sequence editor
- 2) go to its Inspector and select the SMPTE settings tab, where you open the parent item **SMPTE In Chase Lock**
- 3) Once you have accessed the SMPTE IN Chase Lock settings in the inspector (step **2**, graphic above), you can do the following adjustments:

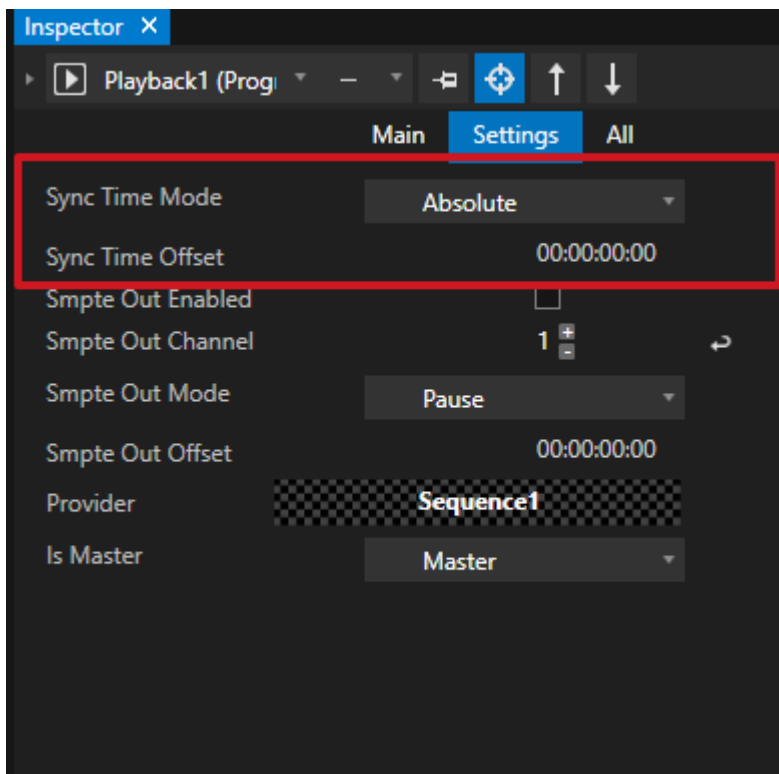
- enable / disable this feature
- select a channel ID corresponding to your interface
- set a timeout delay (in milliseconds)
- enable auto play/ pause to ensure continuous playback after a jump in SMPTE timecode

For legacy support and in special cases do the following to synch the entire system to external SMPTE LTC:

1. Configure your SMPTE-IO Interface  
Set Interface to In, define a Channel ID
2. Select the System where the SMPTE-IO is plugged in the Inspector
3. Check if Inspector Property Mode is set to "Advanced"
4. Go to Settings Tab
5. Set Sync Clock to "SMPTE IO LTC"



6. Go to Project Explorer - navigate to PME Section there
7. Select a Playback as Child of PME LIVE and show Properties into Inspector
8. Set Sync Time Mode of this Playback to "Absolute"
9. Now this Playback is clocked by the Timecode from SMPTE-IO
10. Repeat steps 5-8 for every other Playback you want to lock to the incoming timecode



#### A Playback's Sync Time Mode

The setting to Absolute can come in handy, if the playback needs to be synched to the actual time of the day.

## Temporary Pre-Load

If you need to synchronize VERTEX to a **discontinuous** or **intermittent external timecode** that leads to playback **jumps right into clip containers**, keep in mind that the clip container's **pre-roll time cannot be executed**.

To still ensure that all necessary data is buffered and ready for a smooth playout, create a cue with the script command `TempPreLoad` before the jump. The clip container will be temporarily pre-loaded.

#### Example:

1. The external timecode pauses at 00:01:00.00 then continues at 00:01:30.00
2. In VERTEX a clip container is lined up to start at 00:01:30.00

3. Before the pause at 00:01:00.00, you will need to create a cue with the scripted pre-load command:

```
Playback1.ClipContainer1.TempPreload
```

4. VERTEX will load all necessary data from ClipContainer1 at 00:01:00.00 and have it ready for immediate playback after the jump.

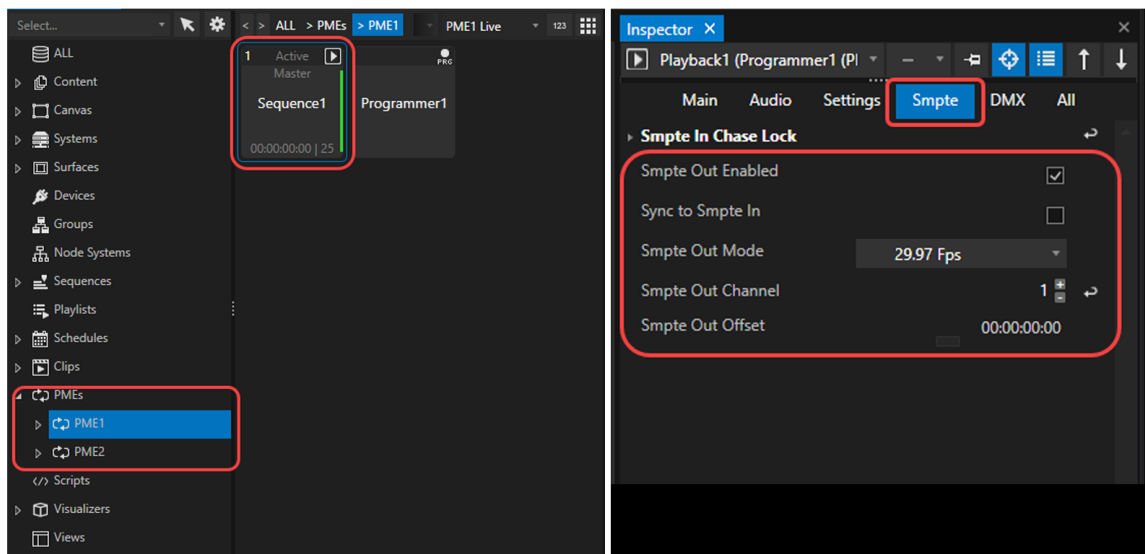
### 6.14.3 Sending Timecode

There are **two options to transmit SMPTE timecode from VERTEX**:

1. Send timecode out from **a playback**
2. Generate SMPTE Timecode **with a SMPTE clip container**

#### Send SMPTE LTC Out From A Playback

Make sure you have [your SMPTE IO interface configured](#).



1

2

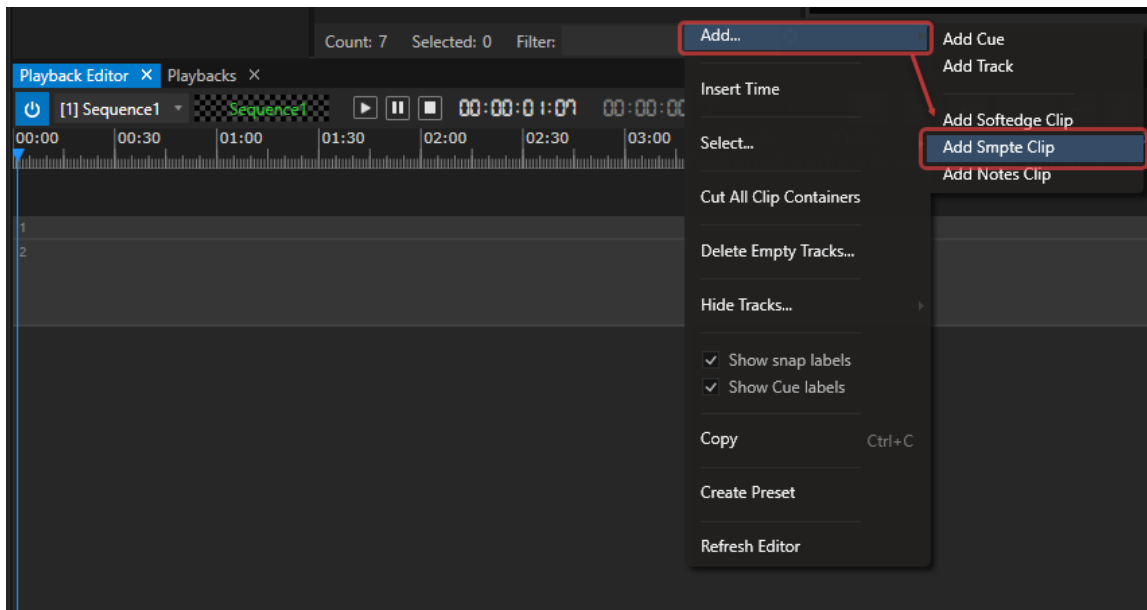
1. go to Project Explorer-> PMEs -> PME1 and select a - for instance Playback1
2. navigate in the Inspector to the SMPTE tab and enable SMPTE Out

Make sure your frame rate in SMPTE Output Mode matches your sequence's and the SMPTE Out Channel matches the one from your interface configuration.

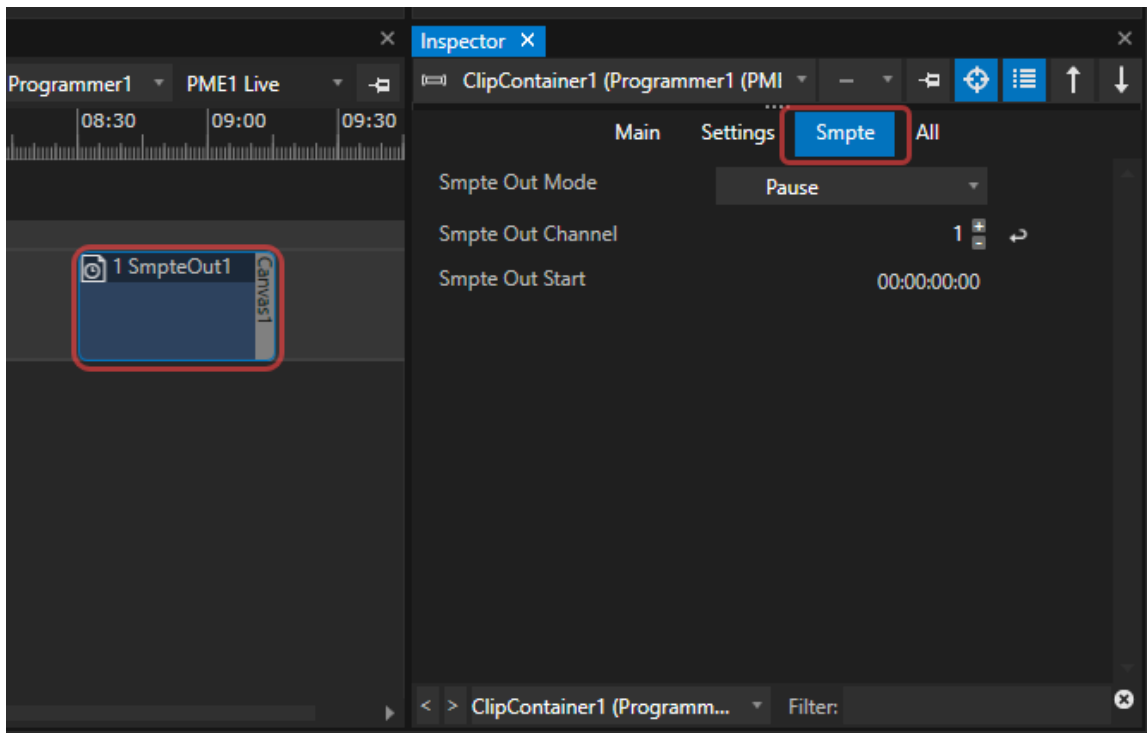


## Generate Timecode With A SMPTE Clip

Generating custom timecode from within VERTEX is easy. There's no need for a third-party application.



1. Plug in and configure your SMPTE IO Interface
2. Go to the Sequence Editor and open the context menu with a right-click on a track.
3. Choose Add SMPTE Clip
4. a new clip container will be created



5. select the clip container and go to the SMPTE tab in its inspector
6. set the start time and output mode
7. the output channel needs to match the channel of your SMPTE-IO interface

## 6.15 Unreal Engine Plugin

- *ioversal provides a **Plugin for Epic's Unreal Engine** that consists of a **Sender and Receiver**.*
- *You are able to **link and control properties** in Unreal and VERTEX **bidirectionally**.*
- *The **Beta Version** is available into the **Download section into your VERTEX user account**: Just log in on [www.ioversal.com](http://www.ioversal.com) or first register there, navigate to "**Support**" or "**My Support**" and select "Downloads" there*

*The documentation for the advanced feature-set of VERTEX is in progress and will be updated step by step.*

*Until then: Please drop us an E-Mail with your "How-to-do-this-in-VERTEX" question to [support@ioversal.com](mailto:support@ioversal.com)*

## 6.16 Vertex Remote Script API

- the **VERTEX Script API** allows you to control VERTEX via third party applications or to built your own service that controls VERTEX
- The **API is based on TCP**
- The script server **listens to all VERTEX Script Commands**

### Details and Settings

**Protocol:** TCP

**Port:** 50009

**IP Address:** Every VERTEX System in your Project has his own Script Server. Choose the IP from that System you want to connect to

**Protocol:** UDP

**Port:** 50019

**IP Address:** Every VERTEX System in your Project has his own Script Server. Choose the IP from that System you want to connect to

You can use all script commands and send them from external over the API to VERTEX

Script Commands **have to be terminated with a carriage return**

VERTEX terminates their answers also with a Carriage return.



#### **Carriage return**

Depends on how your external application works and the carriage return is translated.

Could be \r, CR, or ASCII Code 0D (Hex) or 13(Decimal)

### Examples

- The VERTEX API ist based on the VERTEX own Script Command Language
- Each Script Command that exists in your current VERTEX assembly version could be sent via TCP to the VERTEX API
- Read to [Scripting Topic](#) to learn more about how to write and use VERTEX Script Commands

## Set Parameters

*Opacity for Clip Container 1*

```
ClipContainer1.Opacity.Value = 1\r
```

*Mix Level of Playback1 in PME Live from to full - fade time should be 2 seconds*

```
PME1.Playback1.MixLevel.FadeValue 1,5\r
```

*Set Network Adapter for Art-Net™ on System 1 to "ETHERNET2"*

```
System1.Settings.ArtNetAdapter.Value = ETHERNET2\r
```

*Reset Video Inputs of System 2*

```
System2.ResetVideoInputs\r
```

## Start and stop actions

*Start Playback1*

```
Playback1.Play\r
```

*Stop Playback 3*

```
Playback2.Stop\r
```

*Pause Playback3*

```
Playback3.Pause\r
```

*Run Script 1*

```
Script1\r
```

## Return Values

*Return current Timecode of Playback 1*

```
return Playback1.TimeCode.Value\r
```

Answer from VERTEX:

```
{"TimeSpan":"00:00:00","Days":0,"Hours":0,"Minutes":0,"Seconds":0,"Fra
```

Return a list of all Playbacks in current Project

```
return ListItems.Playback\r
```

Answer from VERTEX e.g.:

```
["Playback1","Playback2"]\r
```

## Change Values

Show notes of clip container 6 in sequence 1 as Text of Textcontent item "Text1"

```
Text1.Settings.Text.Value = Sequence1.ClipContainer6.UserProperties.No
```

Fade Mix Level of Playback1 in PME Live to full - fade time should be 2 seconds

```
PME1.Playback1.MixLevel.FadeValue 1,5 \r5
```

## Advanced Settings

### Script Server Authentication

For each Systems script server, you can activate an authentication by username and password

Both can be set for each System in the Inspector.

When enabled, the script server only will execute scripts, when a valid authentication was sent before.

#### Syntax for Authentication

```
Authenticate:username,password\r
```

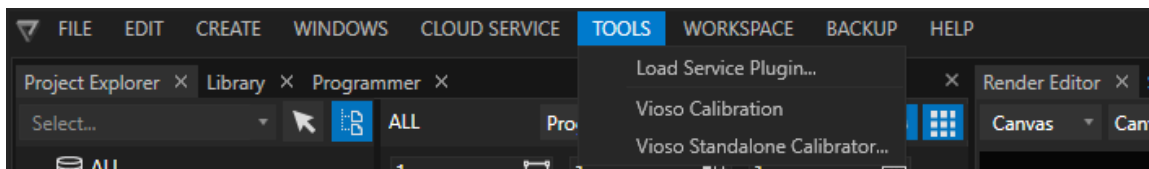
```
e.g.  
Authenticate:Admin,1234\r
```

*Doublecheck, that there is no blank space between the characters!*

## 6.17 VIOSO AutoCal

- VERTEX is shipped with **VIOSO's camera-based calibrator software** for projectors
- A **VIOSO trial** is available for testing in each regular VERTEX version without any additional license key
- VIOSO AutoCal requires an **additional license per output**

### Use VIOSO Autocal



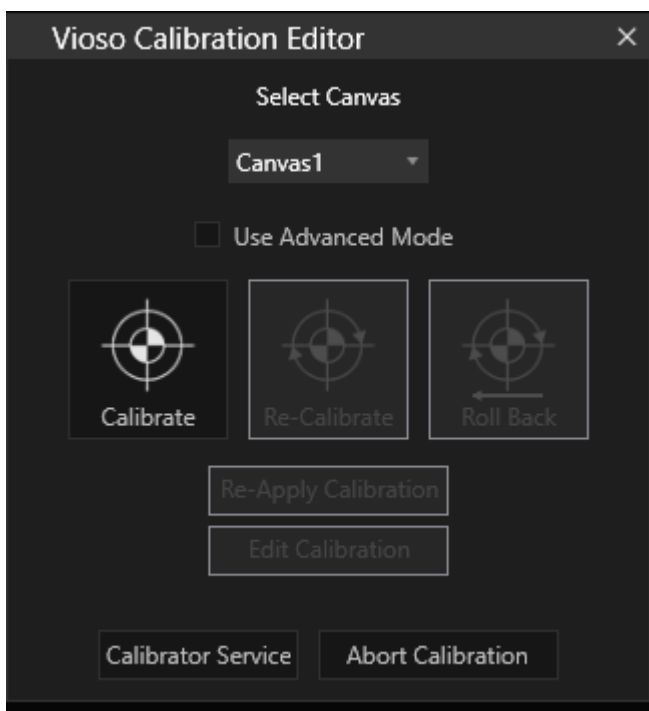
- Go to **TOOLS** section into the **main menu**
- Go to **Vioso Calibration** and open the **Vioso Calibration Editor**





### Vioso Standalone Calibrator

VERTEX comes with a deep integration of VIOSO's camera calibration technology and passes all relevant parameters for surfaces and canvas to VIOSO's calibrator software. Warp and blend files and even VIOSO's own calibration files are handled by VERTEX. For some special needs and projects, it is also possible to start the standalone calibrator. For this very deep knowledge in VIOSO's technology and file structure is strictly recommended



- Select a Canvas - all Surfaces of this Canvas are calibrated with Vioso's calibration software
- For calibration follow the step-by-step guide that is linked below



### Use Advanced Mode

When this checkbox is enabled, there are more options for manual settings during VIOSO's step by step calibration process.

## Licenses

- Based on VIOSOS's licensing model, **AutoCal licenses** are **counted per output**

- To buy a VIOSO license , please contact the [iiversal sales team](#). Also [VIOSO](#) is selling a special VERTEX edition that already is shipped with an AutoCal license.

## Step by step guide



### **VIOSO AutoCal step by step guide**

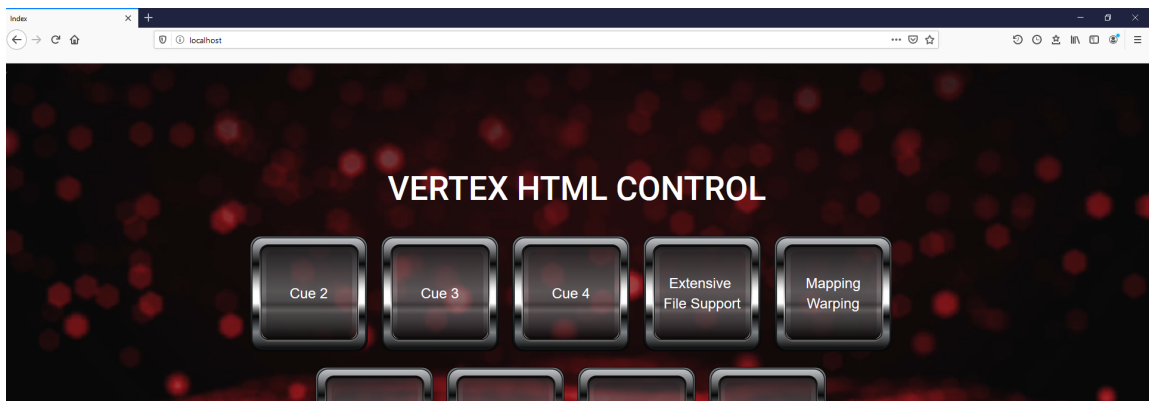
The guys from VIOSO provide a step by step guide for VERTEX. Please follow [this link to their helpdesk](#)

## 6.18 Webserver

### 6.18.1 HTML5 Server Custom Pages

- VERTEX comes up with an **integrated local HTML5 webserver** that can host a **HTML page**.
- Integrate **HTML5 based control panels** and trigger VERTEX with **script commands** with ease.
- To use the webserver **VERTEX has to be started as Windows administrator**.

## Host Websites with Controls



VERTEX comes with a built in webserver that can host HTML5 Content with Javascript and CSS.

With this toolkit on board, you are able to write HTML5 based control panels that can send script commands via a HTML5 form and the post-method.

With the built in webserver you are able to control e.g. a VERTEX project with an tablet computer into the same network.

- HTML-Form
- Method: Post
- Action: Empty
- Element Name = "Script"
- Value = "[ScriptCommand]" eg Value="Playback1.Play"

## Simplified Example

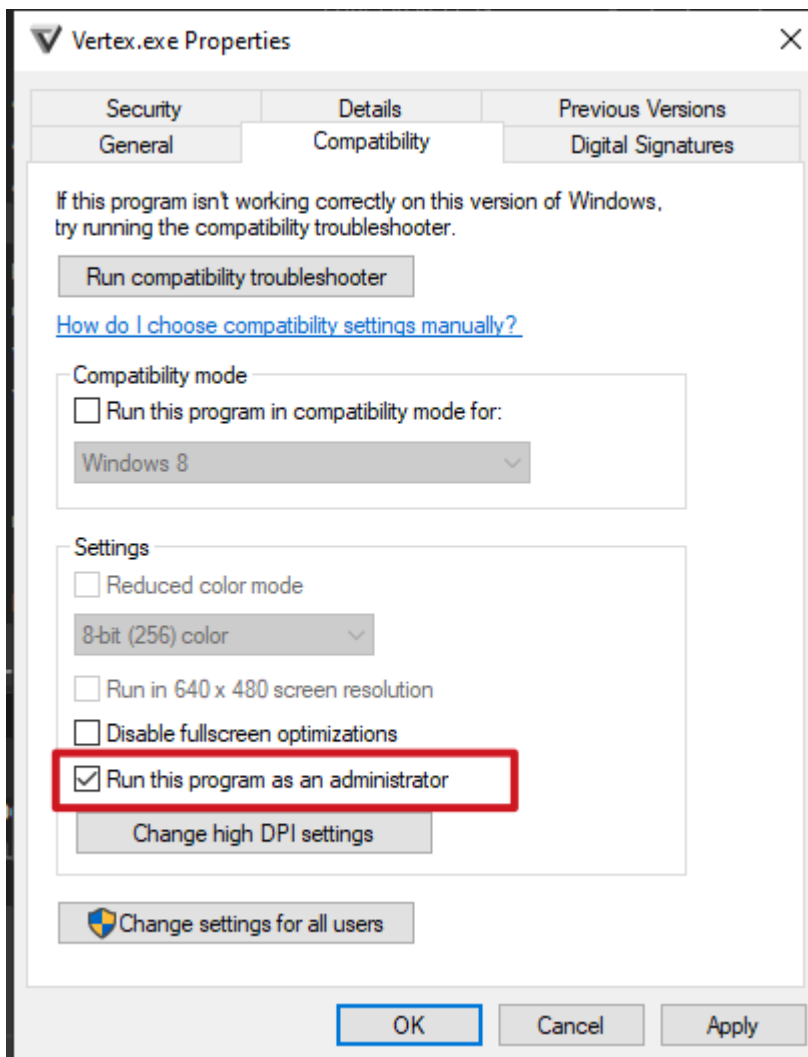
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<form action="/" method="post" target="">
<button name="Script" value="Playback1.Play">Play</button>
</form>
</body>
</html>
```

## Workflow

### 1. Start VERTEX as Admin

*The integrated webserver is the only functionality that requires that **VERTEX as application is started with administrator rights.***

*To permanently start as admin, set the check mark for the Vertex.exe as in the picture below.*



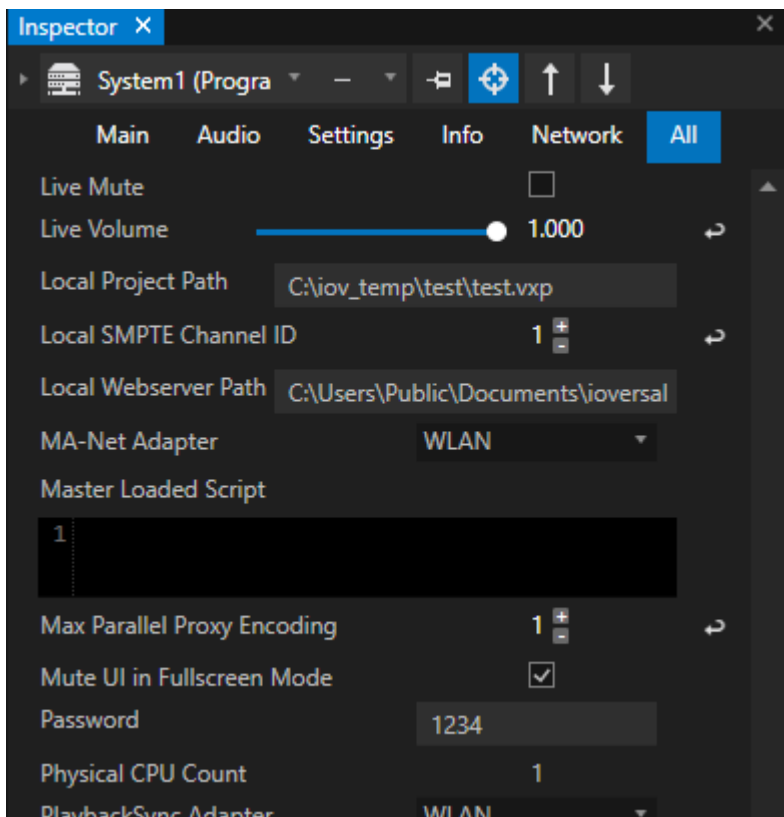
## 2. Set Local Webserver Path for HTML Content

*Set the **local webserver path** into your project and copy your HTML files into this path or use the **standard path** VERTEX sets for every project: C:*

*\Users\Public\Documents\ioversal\Vertex\WebServerData*

*The webserver path can be changed for a system in the inspector.*

*Select "**All**" tab there and scroll to "Local Webserver Path".*



### 3. Copy HTML Files into this Path

*Copy your HTML files into this path.*

### 4. Test with Local Browser

*Enter "localhost" into the address bar of your local webbrowser.*

*The HTML content should be loaded.*

**Firewall-Settings**

*If you plan to access the HTML website from another device in the same network, please set a firewall rule for TCP on port 80.*

*Inbound data receiving has to be allowed on this port.*

*Misc.*



## 7 Misc.

- *Collection point for helpful topics, additional information and miscellaneous issues*

### List of Script Commands

*A list of all Script Commands*

### VERTEX Data Formats and File Suffixes

*Get an overview of all file formats that are used in or created by VERTEX*

### Keyboard Shortcuts

*Information about access to keyboard shortcuts*

## 7.1 List of Script Commands

Find below a list of the most common scripting commands available for VERTEX CORE objects as a reference.

### Jump to Categories:

- [Top/Root Level](#)
- [General Property Scripts](#)
- [System](#)
- [Systems Manager](#)
- [Backup Service](#)
- [Clip Container](#)
- [Programmer](#)
- [PME Playback](#)
- [Content](#)
- [Webbrowser Content](#)
- [Control View](#)
- [Devices](#)



### Show a list of all available script commands.

VERTEX comes with a dynamic script languages. Script commands are GUI-oriented in regards to the workflow.

**To show a list of all available script commands**, just click one of the script fields or into command section at the bottom and **press "CTRL and Space"**. A list of all available commands opens. Select an item from the list and press ENTER.

To show **all available commands for the next deeper level**, enter a decimal point "."

## TOP/ROOT LEVEL

**ProjectClose**

**ProjectLoad**

**ProjectSave**

**ProjectSaveBackup**

**ProjectLoadPreviousBackup**

**ProjectLoadPreviousAutoSave**

**ProjectSettings****ViosoCalibrate****ViosoCalibrateAdv****ViosoReCalibrate****ViosoReApply****MainWindowMinimize****MainWindowMaximize****MainWindowExit**

This will immediately exit & close VERTEX without saving.

**Wait**

Wait for a specified amount of time. Parameters: seconds (double)

**Return**

Stop script execution and return the specified result. Parameters: result

**Log**

Write to the script monitor's output window. Parameters: message

**Clear**

Clear the script monitor's output window.

**WaitAll**

Wait for all executing child scripts to finish. Parameters: timeout/seconds (double)

**ListAllItems**

Return a list of all scriptable (top-level) items.

**Cancel**

Cancel all running scripts.

**ListItems**

Returns a list of all items of a specific type.

## Conditional Script Commands

**IfEqual**

Executes the specified script if  $a = b$ . Parameters: a, b, script

**IfEqualEqual**

Executes the specified script if  $a = b$  and  $c = d$ . Parameters:  $a, b, c, d$ , script

**IfEqualGreater**

Executes the specified script if  $a = b$  and  $c > d$ . Parameters:  $a, b, c, d$ , script

**IfEqualLesser**

Executes the specified script if  $a = b$  and  $c < d$ . Parameters:  $a, b, c, d$ , script

**IfEqualUnequal**

Executes the specified script if  $a = b$  and  $c \neq d$ . Parameters:  $a, b, c, d$ , script

**IfExecute**

Executes a specified script if a local variable has a Boolean value of 1 or true

**IfGoto**

[Jumps to a tag](#) if the condition - the value of the local variable - is true.

**IfGreater**

Executes the specified script if  $a > b$ . Parameters:  $a, b$ , script

**IfGreaterGreater**

Executes the specified script if  $a > b$  and  $c > d$ . Parameters:  $a, b, c, d$ , script

**IfGreaterLesser**

Executes the specified script if  $a > b$  and  $c < d$ . Parameters:  $a, b, c, d$ , script

**IfLesser**

Executes the specified script if  $a < b$ . Parameters:  $a, b$ , script

**IfLesserLesser**

Executes the specified script if  $a < b$  and  $c < d$ . Parameters:  $a, b, c, d$ , script

**IfUnequal**

Executes the specified script if  $a \neq b$ . Parameters:  $a, b$ , script

**IfUnequalGreater**

Executes the specified script if  $a \neq b$  and  $c > d$ . Parameters:  $a, b, c, d$ , script

**IfUnequalLesser**

Executes the specified script if  $a \neq b$  and  $c < d$ . Parameters:  $a, b, c, d$ , script

**IfUnequalUnequal**

Executes the specified script if  $a \neq b$  and  $c \neq d$ . Parameters:  $a, b, c, d$ , script

**PopupNotification**

Create a popup notification on top of the user interface. Usage: `PopupNotification WarningMessage`

**PopupNotificationWithClose**

Create a popup notification with close button on top of the user interface. Usage: `PopupNotificationWithClose`  
`WarningMessage`

## GENERAL PROPERTY Scripts

### ***ProgValue***

Get/set the property's value in the current programmer.

### ***FadeProgValue***

Fade the property's value in the current programmer. Parameters: value (double), seconds (double)

### ***SetProgRelative***

Add a relative value to the property's value in the current programmer.

### ***Value***

Get/set the property's value.

### ***FadeValue***

Fade the property's value. Parameters: value (double), seconds (double)

### ***SetRelative***

Add a relative value to the property's value.

## SYSTEM

***EnterFullScreen***

***LeaveFullScreen***

***FullScreenToFront***

***SmpteloSetMode***

***SmpteloSetTime***

***RunProcess***

***RunProcessArgs***

***KillProcess***

***WindowsShutdown***

***WindowsRestart***

*ResetVideoInputs*  
*ForceResync*  
*ResetAsioDevice*  
*MouseMove*  
*MouseLeftClick*  
*StartVertex*  
*StopVertex*  
*ReconnectTimeServers*  
*RefreshNetworkAdapterList*  
*Connect*  
*Disconnect*

*NotchPurgeVRam*  
*RefreshDmxOutputs*  
*GetStatus*  
*LogStatus*  
*WriteToLog*

#### ***ScriptServerSendMessage***

*Send message as string. Byte or hex codes can be included like this {{72 101 108 108 111 32 0x57 0x6f 0x72 0x6c 0x64 0x13}}*

#### ***ScriptServerSendMessageToIp***

*Send message as string. Byte or hex codes can be included like this {{72 101 108 108 111 32 0x57 0x6f 0x72 0x6c 0x64 0x13}}*

## SYSTEMS MANAGER

*GetStatus*  
*AllSessionMembersEnterFullScreen*  
*AllSessionMembersLeaveFullScreen*

## BACKUP SERVICE

*Reconnect*  
*Separate*  
*TakeOverMasterRole*

## CLIP CONTAINER

*TempPreload*

*TempUnload*

## PROGRAMMER

*StoreAllData*

*ClearAllData*

## PME / PLAYBACK

**GotoCue**

*Transport playback to the specified cue. Parameters: cue number (int)*

**GotoCuePlay**

*Transport playback to specified cue and switches to Play. (Useful for Pause Cues)*

**LoadCue**

**ReleaseCue**

**GotoTime**

*Set Transport playback to the specified time.*

**GotoFrame**

*Transport playback to the specified frame.*

**IgnoreNextCue**

**EnableAllCues**

**GotoPrevCue**

**GotoNextCue**

**GotoFirstCue**

**GotoLastCue**

**GotoPreviousFrame**

**GotoNextFrame**

**Pause**

**Play**

**Stop**

**TogglePlay**

*GetPlaybackTime*  
*GetPlaybackCurrentCueTime*  
*GetPlaybackRemainingCueTime*  
*GetPlaybackTransport*  
*GetCurrentCue*  
*GetNextCue*

*FadeToCue*  
*FadeToCueHold*  
*FadeToCueHoldPlay*  
*FadeToCuePlay*  
*FadeToNexCue*  
*FadeToNexCuePlay*  
*FadeToPreviousCue*  
*FadeToPreviousCuePlay*  
*FadeToTime*  
*FadeToTimeHold*  
*FadeToTimeHoldPlay*  
*FadeToTimePlay*

## CONTENT

*SetVersion*  
*SetNextVersion*  
*SetPreviousVersion*  
*SetNextCycleVersion*  
*SetPreviousCycleVersion*

## WEBBROWSER CONTENT

***NavigateTo***  
*Navigate to new Url.*

***Back***  
*Navigate back.*

***Forward***



*Navigate forward.*

**Reload**

*Reload current Url.*

**ReloadCache**

*Reload current Url and Cache.*

**SendKey**

*Send Key: Left Right Up Down PageUp PageDown.*

**SendJs**

*Send JavaScript Code to current page.*

## CONTROL VIEW

**Close**

**Open**

**MoveBackward**

**MoveForward**

**GotoPage**

*Navigates to the specified page within the current control view. Parameters: page name (string), transition duration (seconds, optional)*

**SetView**

*Navigates to the specified view (and page). Parameters: view name (string), page name (string, optional), transition duration (seconds, optional)*

**GetPage**

**GetView**

## DEVICES

## General Device Commands:

*Connect*

*Disconnect*

## StreamDeck:

*SetKeyColor*

## UDP Sender:

*ResetConnection*

*SendMessage*

## TCP Client:

*ResetConnection*

*SendMessage*

## TCP Connection:

*ResetConnection*

*Resets the connection.*

*SendMessage*

*Sends a string message to all connected endpoints. Byte or Hex Codes can be included like this {72 101 108 108 111 32 0x57 0x6f 0x72 0x6c 0x64 0x13} Use \{ or \} to send curly brackets.*

*SendMessageToIp*

*Sends a string message to a specific endpoints.*

## Process Device:

*Start Start the process.*

*Stop Stop the process.*

*Maximize*

*Maximize the process main window.*

**Minimize**

*Minimize the process main window.*

**Restore**

*Restore the process main window.*

**PPT Device:**

**ResetConnection**

**CloseConnection**

**NextSlide**

**PreviousSlide**

**GotoSlide**

**SetVolume**

**Launch**

**Quit**

**StartPresentation**

**EndPresentation**

**OSC Sender:**

**ResetConnection**

*Resets the connection.*

**SendMessage**

*Sends an Osc message with one or multiple value. Usage: Address,Value,Value,...*

**SendBundle**

*Sends an Osc Bundle with one or multiple values. Usage: Address,Value,Value,...*

**KNX:**

**ResetConnection**

**SendTrue**

**SendFalse**

**SendDpt9**

**SendDpt14**

**Kiosk Browser:**

**Start**  
**Stop**  
**Maximize**  
**Minimize**  
**Restore,**  
**NavigateTo**  
**Back**  
**Forward**  
**Home**  
**Reload**  
**ReloadCache**

## Examples

*Start Playback1*

```
Playback1.Play
```

*Stop Playback 3*

```
Playback2.Stop
```

*Pause Playback3*

```
Playback3.Pause
```

*Run Script 1*

```
Script1
```

*Show the notes of clip container 6 from sequence1 as text of text-content item "Text1"*

```
Text1.Settings.Text.Value = Sequence1.ClipContainer6.UserProperties.No
```

*Fade mix level of playback1 in PME live to full - fade time shall be 2 seconds*

```
pme1.Playback1.MixLevel.FadeValue 1,5
```

*Set opacity for clip container 1 to value 1*

```
ClipContainer1.Opacity.Value = 1
```

*Set background color of clip container 1 to color red =0.5, blue= 1, green = 1, alpha = 1*

```
Sequence1.ClipContainer1.BackgroundColor.Value = 0.5,1,1,1 //normalize
```

*Fade mix level of playback1 in PME live from to full - fade time should be 2 seconds*

```
PME1.Playback1.MixLevel.FadeValue 1,5
```

*Set network adapter for Art-Net™ on system 1 to "ETHERNET2"*

```
System1.Settings.ArtNetAdapter.Value = ETHERNET2
```

*Reset video inputs of system 2*

```
System2.ResetVideoInputs
```

*Reset video inputs of system 2.*

```
System2.ResetVideoInputs
```

*Change the label text of label 1 in control view 1.*

```
ControlView1.Controls.Label1.Settings.Caption.Text.Value = "this is a
```

*Perform a click on button 1 of control view 1.*

```
ControlView1.Controls.ClickButton1.Click
```

*Return the current page that is displayed by control viewer 1.*

```
ControlViewer.GetPage
```

Switch to page 2 of the current control view. Also works as script for e.g. button in control view editor (run mode).

```
ControlViewer.GotoPage Page2
```

Delete label 1 on page 1 of control view1.

```
ControlView1.Pages.Page1.Label1.Delete
```

Assign content to a clip container:

```
Playback1.ClipContainer1.MainContent.Value = Content1
```

## 7.2 VERTEX Data Formats and File Suffixes

Quick Overview about the data formats and file extensions that are used in or created by VERTEX:

File extension	Usage	Description
.vxp	<b>VERTEX Project File</b>	Your VERTEX project is saved as Project File by default. This file includes all of your project data and settings
.vxdi	<b>VERTEX Device</b>	This file contains all information and possible control settings about an 3rd Party device, e.g. Matrix switcher, Projectors or DMX Devices
.vxdx	<b>VERTEX Device (encrypted)</b>	Same like .vxdi, but encrypted and protected. Library elements, created by ioversal, that are shipped with VERTEX are encrypted and protected against Copy.
.vas	<b>VERTEX Autosave File</b>	Project File that is located into subfolder "_bak" into your Project Folder. VERTEX autosaves Project Files with this file suffix. To recover a Project File, rename this File to .vxp again or just open the file with VERTEX.
.vwf	<b>Vioso Warping File</b>	This file is exported after a successful processed Vioso Calibration. It contains all warping information for a Surface. For each of your selected Surfaces of a Canvas, one of this Vioso Warping Files should be generated.
.sps	<b>Vioso Project File</b>	The Project file contains all information about displays and warping of the Vioso Calibration tool. This kind of files could have file sizes of many Gigabytes because all calibration data is stored there. For editing or recalibrate an existing calibration, the Vioso AutoCal Tool will need this kind of files.
.iovi	<b>ioversal image sequence format</b>	Proprietary format for image sequences. Selectable option during image sequence import. Keeps all data from original file format but saves the picture information in an optimized internal order to guaranty the best performance for image sequence playback





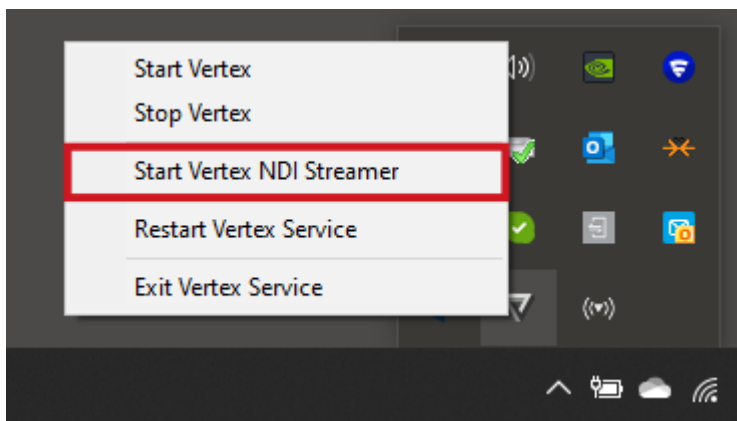
*VERTEX NDI Streamer*

## 8 VERTEX NDI Streamer

- VERTEX NDI Streamer is a **multi source NDI toolkit**
- VERTEX NDI Streamer is **capable to stream up to 2 desktop regions and 1 live input** as a NDI Stream
- The NDI Streamer **is free** and fully **works without a valid VERTEX license**
- **In addition, all settings and streams can be controlled remotely out of VERTEX (with a license)**

### Start NDI Streamer

- The VERTEX NDI Streamer comes up in a package with the VERTEX installer
- Just download and install an up-to-date Version of VERTEX
- Right-click with your Mouse on the VERTEX Tray Icon on the Windows taskbar
- Start Vertex NDI Streamer

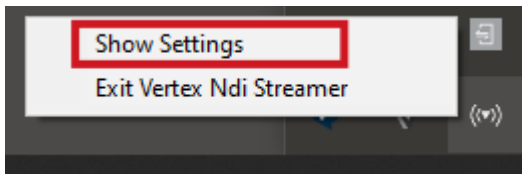


- When the NDI Streamer is started, a new Icon appears into the windows tray bar



## Set up NDI Streams

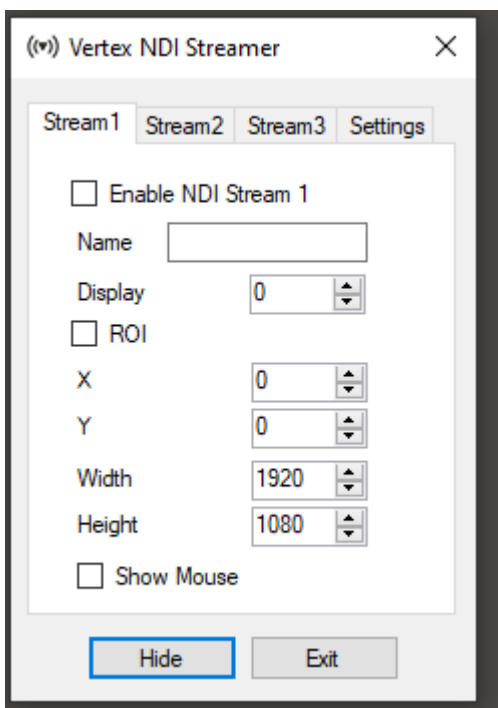
- Right-Click on the VERTEX NDI Streamer tray icon in the windows taskbar
- Select "Show Settings"



- Now you can enable up to 3 NDI Streams from your PC

### Stream 1 + 2

enable you to set up two NDI streams that capture a display (or only a region) of your PC.



**Name**

Name of the stream that is shown for all NDI applications

**Display**

Select the display number of your PC that should be sent as NDI stream.

**ROI**

Region of Interest. Default: disabled

Enable to capture only a region of the selected display as NDI stream

Enter pixel coordinates (starting at top left) and width/height to define the ROI

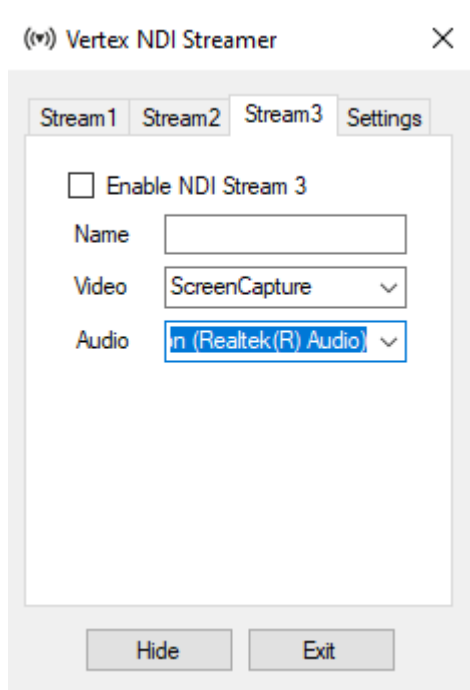
**Show Mouse**

Show mouse pointer in NDI stream

## Stream 3

Stream 3 differs from the first ones. Here, you are able to send a video source (e.g. a webcam) and audio source (like a microphone)

from your Windows PC as NDI stream.



**Name**

Name of the stream that is shown for all NDI applications

**Video**

Select from a list of video sources.

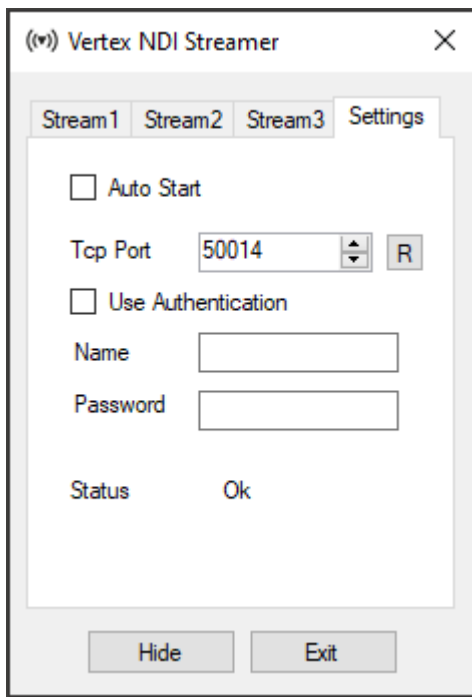
Depending on your PC hardware and configuration the list of available sources may vary

**Audio**

Select from a list of audio sources.

Depending on your PC hardware and configuration the list of available sources may vary

**Settings**



**Auto Start**

Enable to add the NDI Streamer to Microsoft Windows autostart

**TCP Port**

Change port number for remote controlling the NDI streamer over TCP.

Use the R-Button to restart the TCP Servers after a port change.

**R Button**

Reset and Restart the TCP Server for all network adapters after you have changed the TCP Port

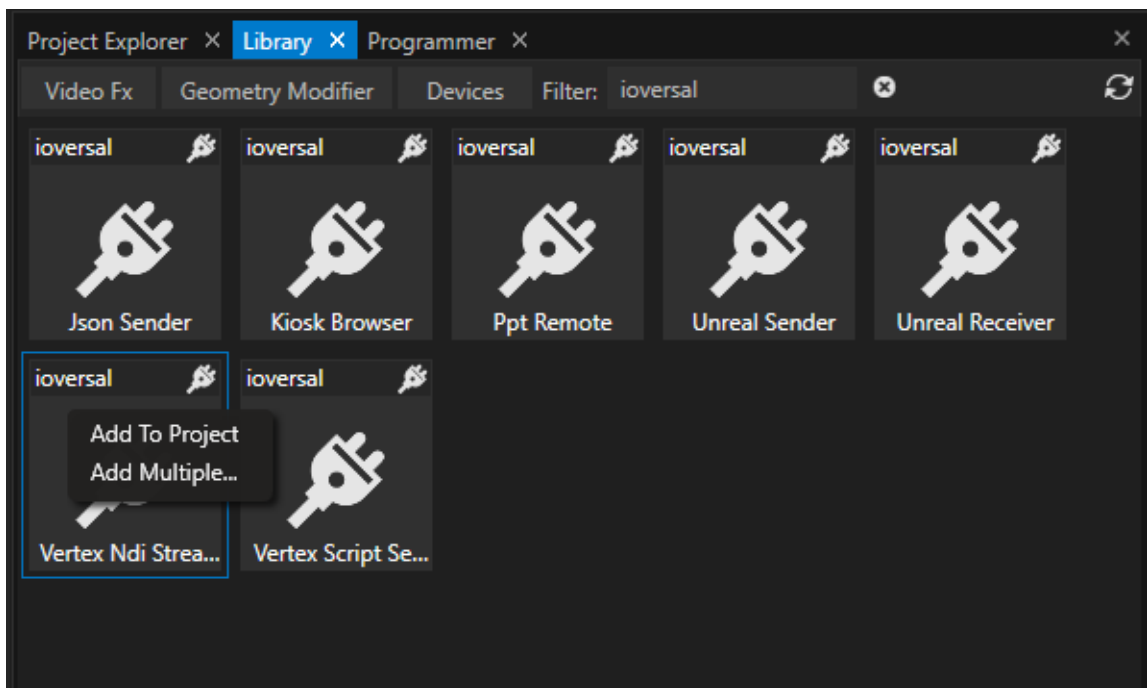
**Use Authentication**

Set User Name and Password for TCP Authentication

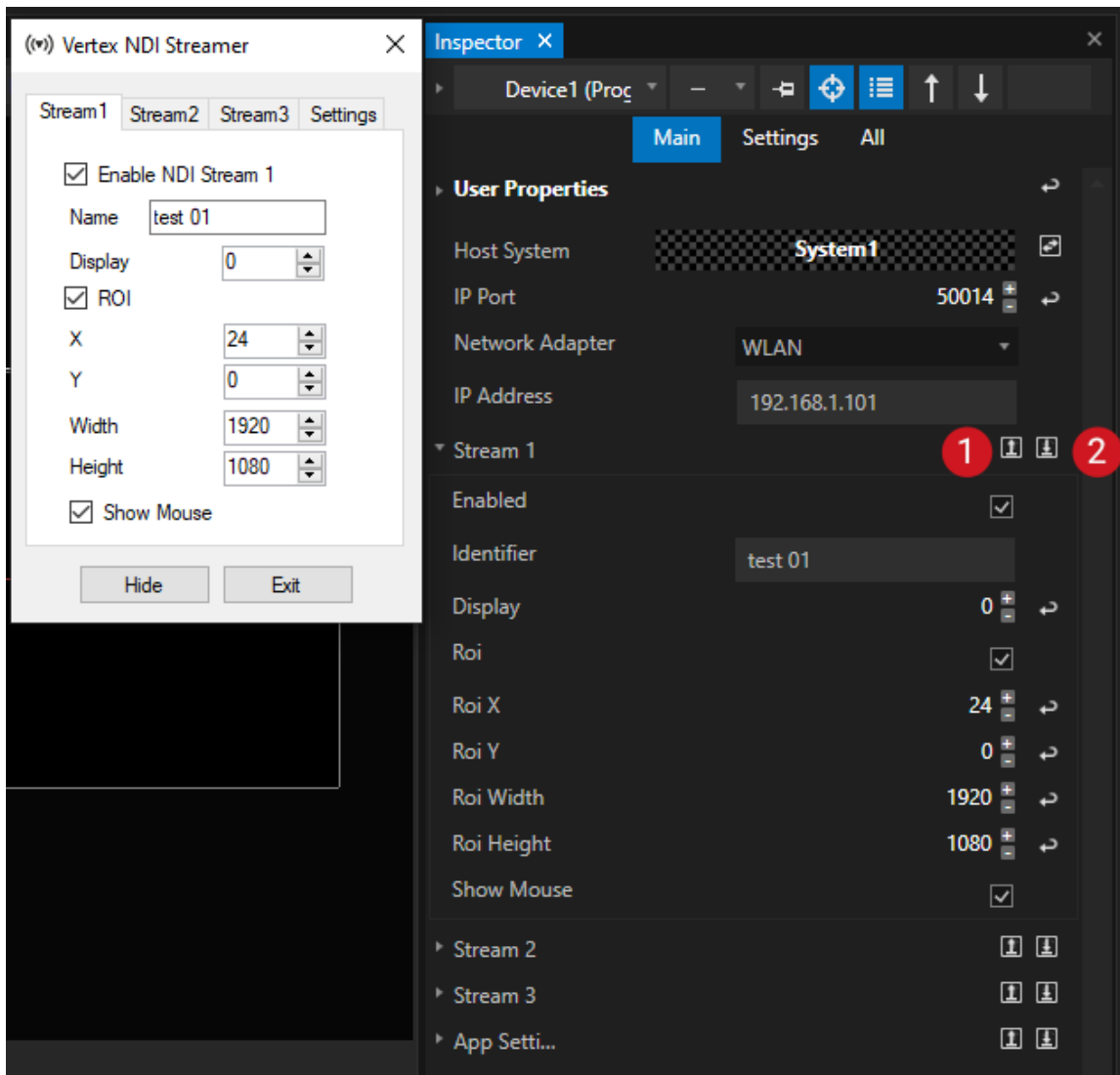
## Remote Control out of VERTEX

- You can **remote control all settings of the VERTEX NDI Streamer on a PC in your network out of VERTEX**
- Just add a **"Vertex NDI Streamer" Device** from the [Library](#) into your current project
- Once configured, you have **access to all settings of the NDI Streamer application** and are able to automate your NDI workflow

### Setup



- Open the [Library](#)
- Select **Devices** and filter for **"ioversal"**
- Right-click to the device **"Vertex NDI Streamer"**
- **"Add to project"**
- Select the device in the [Project Explorer](#) and set the initial properties in the [Inspector](#)
- Once the connection is set up, you are able to upload and download all application and stream settings.



Comparison: All settings from the NDI Streamer application (on the left) you will also find into the Inspector when selecting the corresponding Device "Vertex NDI Streamer" (on the right)

**IP Port**

Enter here the same IP Port as set in the NDI Streamer application

**Network Adapter**

Choose a network adapter from your local VERTEX instance

**IP Address**

Enter IP address of the PC on that the Vertex NDI Streamer application is running and should be remote controlled

**Upload Settings** 1

Upload all current settings from VERTEX to the NDI Streamer application

### Download Settings 2

Download all current settings from NDI Streamer application to Inspector

## Scripting

As with every Device in VERTEX, you can control an NDI Streamer Device with a [Script](#).  
For the following examples the NDI-Streamer Device was added as "Device1" to a project

### Enable and disable Stream 1 on the remote PC

```
Device1.Settings.Stream1.Enabled.Value = true  
Device1.Settings.Stream1.Enabled.Value = false
```

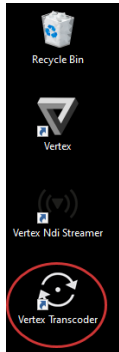
### Show Mouse Cursor in NDI Stream 1

```
Device1.Settings.Stream1.ShowMouse.Value = true
```



*VERTEX Transcoder*

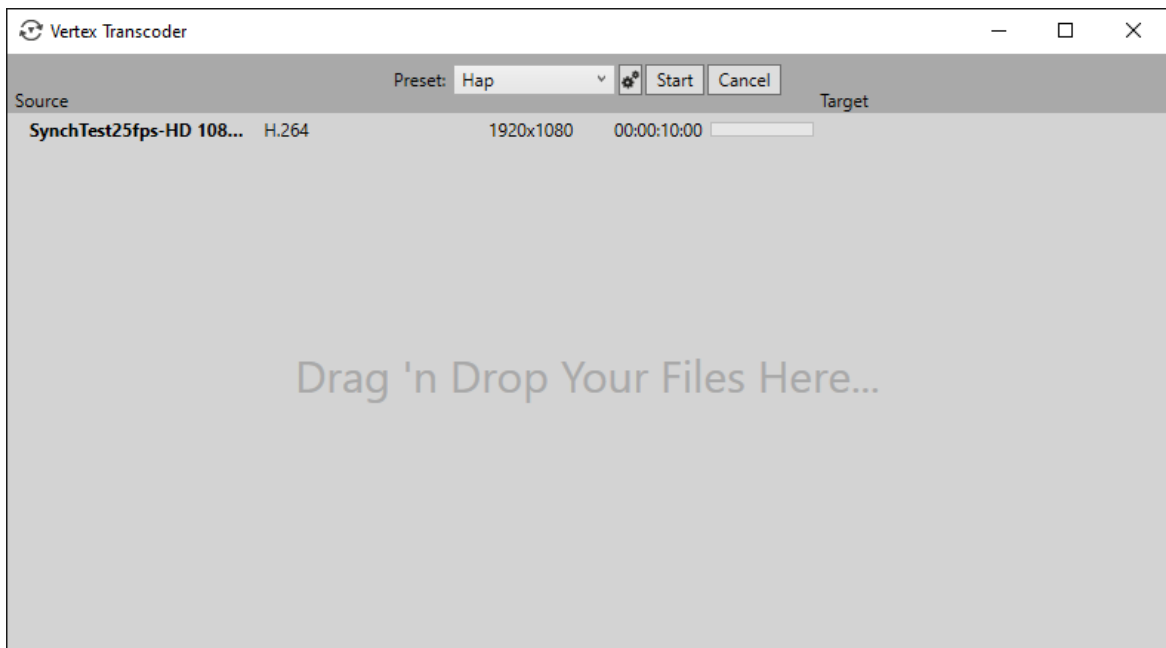
## 9 VERTEX Transcoder



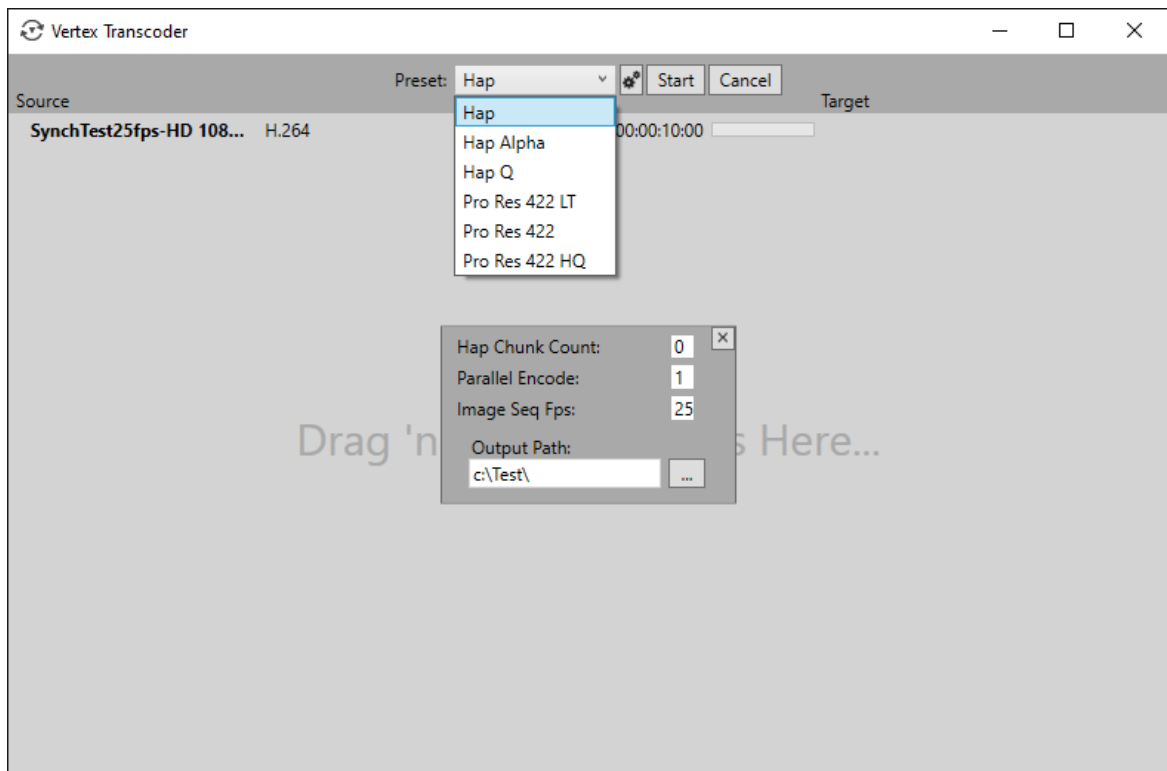
- VERTEX comes with a separate application for video transcoding. Here's why:
- While similar to [transcoding from within VERTEX](#), the separate app offers some more options.
- Also, at times when VERTEX itself is using most systems resources for complex tasks, it can be beneficial to outsource transcoding large chunks of video to another app or system.
- Transcoder is located in VERTEX program folder. The installer creates a shortcut of the Transcoder app and places it in the desktop folder.
- transcode to various HAP and Pro Res codecs (HAP, HAP Q, HAP Alpha, Pro Res 422 including LT & HQ)

### Ease Of Use

Open VERTEX Transcoder from your desktop. And just drag and drop video files into VERTEX Transcoder, choose the transcoding preset from the drop-down menu and click START.

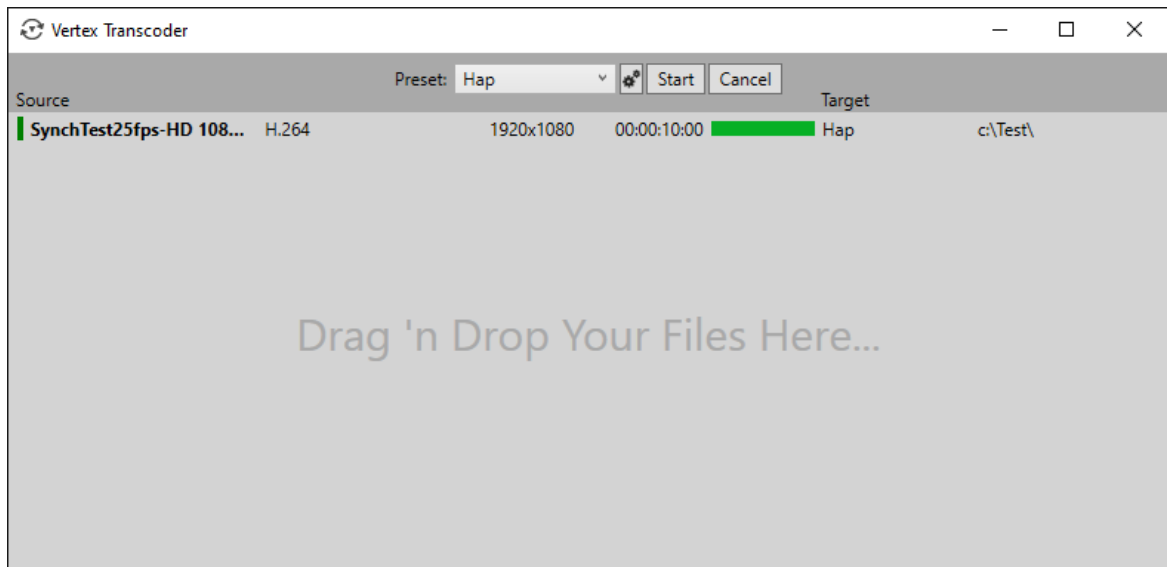


A click on the gear icon on the top bar reveals additional settings:



- **Hap Chunk Count:** When chopping each frame of the movie into separate chunks, please keep in mind that the number of chunks should not exceed the number of the rendering system's processor cores.
- **Parallel Encode:** less parallel threads lead to lesser latency, but performance will suffer slightly and vice versa.
- **Image Seq Fps:** set the frames per second when transcoding an image sequence
- **Output Path:** choose the target directory

A green progressive status bar accompanies the transcoding process:



**- . -**

.3DS 238  
 .FBX 238  
 .iovi 233  
 .mov 223  
 .pptx 193, 476  
 .psd 246  
 .vas 288  
 .vxdi 173, 410, 456  
 .vxdx 173, 410, 456

**- \_ -**

\_bak 288

**- 1 -**

10 Bit 233  
 10Bit Playback 223  
 16 Bit 233

**- 3 -**

3D Object 128  
 3D Objects 238  
 3DS 238

**- 6 -**

64 Bit 23

**- 8 -**

8 Bit 233

**- A -**

about Vertex 9  
 Absolute Keyframe 145  
 Access a Script 655  
 Active Values 405, 413  
 Add Content 213  
 Add Devices 173, 456  
 Add Live Input 246  
 Add Output 330

add surface 70  
 Add Surface to Canvas 69  
 Add Track 122  
 Adobe Photoshop 246  
 Advanced 392  
 Advanced Mode 367, 583  
 Advanced Scalling Settings 330  
 Adveanced Mode 392  
 AIA 246  
 All Output 80  
 AMD Eyefinity 23  
 API 684  
 Application Programing Interface 684  
 Areas 269, 270  
 ArtNet 321, 496  
 Art-Net 181, 185, 464, 468, 496, 509, 542  
 ASIO 96, 169, 354  
 ASIO Audio 96, 354  
 ASIO Output Channel Count 96  
 ASIO Thread Priority Highest 96  
 Assembly Version 292  
 Audio 104, 223, 238, 334, 354  
 Audio Channel Map 122  
 Audio Channel Mapping 96  
 Audio Channel Routing 96  
 Audio Clip 128  
 Audio Clock 96  
 Audio Extraction 287  
 Audio Import 238  
 Audio Offset 113, 152, 169  
 Audio Output 96  
 Audio Output Channels 96  
 Audio Outputs 69  
 Audio Playback 334, 353  
 Audio Preview 96  
 Audio Routing 169  
 Audio System 96, 104, 169  
 Authentication 684  
 Authorization 15  
 Auto Content Sync 311  
 Auto Fade 113  
 Auto Start Settings 419  
 Auto Suggestion 655  
 AutoCal 84, 688  
 AutoCal Plus 688  
 Autoconnect 293  
 AutoFade 145  
 Autosave 288  
 Autosave Interval 288  
 Auto-Star File 419

Autostart 206, 419, 489, 714  
 Auto-Start Folder 419  
 Auxiliary textures 239  
 Available Systems 293

## - B -

Background 444  
 Background Service 26  
 Backup 20, 292, 322  
 Backup Group 292, 322  
 Backup Master 322  
 Backup Mode 322  
 Backup Service 633, 698  
 Backup Sync Clock Source System 307  
 Backup System Clock 322  
 Basics 14, 67  
 Bezier 145  
 Blackmagic Design 246  
 blend file 69, 84, 88, 688  
 Bluefish444 246  
 Blueprint 683  
 BMP 233, 238  
 Bookmarks 274  
 brightness 421  
 Browser 260  
 Button 444  
 Buttons 15, 444  
 Buy Licenses 33

## - C -

C++ Redistributables 26  
 Cached Image Sequence 233  
 Calculation 563  
 Calender 616  
 Camera 246, 377  
 Camera Calibration 688  
 Cancel 645  
 Canvas 67, 69, 88, 102, 104, 113, 128, 377  
 Canvas Model 104  
 Canvas Offset 69, 88, 104  
 Canvas Size 104  
 Canvas Space 67, 69, 102, 104  
 Canvas View 377  
 Capture Cards 246  
 Carriage Return 684  
 ChamSys 425, 509  
 Change Version 280

Changelog 62  
 Channel Map 169, 425, 509  
 Channel Routing 96  
 Clear Log 645  
 Clear Programmer 413  
 Client 19  
 Clip 128, 152, 166, 383  
 Clip Conatiner 425, 509  
 Clip Container 113, 122, 128, 145, 166, 168, 169, 185, 239, 270, 277, 383, 468, 542, 583, 633, 698  
 Clip Container Icons 128  
 Clip Container Settings 128  
 Clip Target 128  
 ClipContainer 633, 698  
 Codecs 223  
 Collections 214, 269, 274, 369  
 color correction 421  
 color temperature 423  
 Colors 619  
 Command Systax 619  
 Composite 604  
 Composite Node 559, 583, 604  
 Compositing 113  
 Compressed Cache 233  
 Condition 556, 559, 583  
 Conditions 619  
 Condtitions 595  
 Configuration 675  
 Configure AutoFade 145  
 Configure Outputs 69  
 Connect 293  
 Connect a Node 583  
 Connect to IP 293  
 Connected Input 583, 595  
 Connection 583  
 Connection line 583  
 Connectivity 173, 456  
 Console 645  
 Console Layer 128, 425, 509  
 ConsoleLayer 496  
 Content 166, 214, 223, 280, 282, 311, 369, 633, 698  
 Content Encryption 223  
 Content Management 280  
 Content Sharing 311  
 Content Sync 20, 292, 311  
 Content Transfer Monitor 311  
 Content Types 213  
 Content Versioning 269, 280  
 Control Panel 444  
 Control View 633, 698

Control Viewer 367, 444  
 Control with DMX-512 509  
 ControlView 444  
 ControlView Editor 444  
 ControlViewer 444  
 Copy 383  
 Core Elements 369  
 Count 578  
 CPU 23, 563  
 CPU load 367  
 Create a Node 583  
 Create Content 214  
 Create Group 277  
 Create Nested Sequence Content 113  
 Cue 149, 152, 166, 383  
 Cue ID 149  
 Custom Name 270  
 Custom Nodes 583

## - D -

Damping 551  
 DANTE 96  
 Data Formats 711  
 Data Monitoring 570  
 Data Routing 20, 292, 321  
 Data Sync 20, 266, 292, 306  
 Data Sync Adapter 266, 306  
 Datapath 246  
 Date 616  
 Deactivate 583  
 Deactivation 33  
 Debug Script Commands 645  
 Decimal 496  
 Decompose 604  
 Default Canvas 113, 152  
 Default Datasync Adapter 293  
 Default Workspace 358  
 Delay 152  
 Delete Outputs 330  
 DeltaCast 246  
 Deltacast FLEX 246  
 Desktop Capture 246  
 Device 128, 173, 181, 193, 200, 410, 456, 464, 476, 483, 509  
 Device Path 173, 410, 456  
 Devices 173, 185, 206, 277, 321, 410, 456, 468, 489, 542, 563, 583, 633, 698, 714  
 Digital Rights Management 223  
 Dimmer 185, 468, 542  
 DirectX 11 23  
 Disconnect a Node 583  
 Display Surface 67, 69  
 Disssconnect 293  
 DMX 496  
 DMX 512 181, 464, 496, 509  
 DMX Channels 425, 509  
 DMX Control 509  
 DMX Devices 185, 410, 468, 542  
 DMX In 507  
 DMX Input 181, 425, 464, 496, 509  
 DMX Input enabled 509  
 DMX Interface 496, 507  
 DMX IO 507  
 DMX Mode 425, 509  
 DMX Out 185, 468, 507, 542  
 DMX Output 496  
 DMX over Ethernet 496  
 DMX over Network 496  
 DMX Patch 496  
 DMX Routing 185, 468, 542  
 DMX Routing Editor 181, 464, 496, 509  
 DMX Start Address 509  
 DMX Universe 509  
 DMX-512 185, 321, 425, 468, 496, 509, 542  
 DMX-512 Input 509  
 DMX-512: Control a Console Layer 509  
 DMX-512: Control a Playback 509  
 DMX-512: Control a Surface 509  
 DMX-512: Control a System 509  
 DMX-Input-Device 509  
 DMX-IO 496, 507  
 DMX-IO Configuration 507  
 DMX-Routing 496, 509  
 DMX-Routing-Editor 496  
 Dongle 15, 33, 223  
 Dongle lock 15  
 Dongle Offline Activation 33  
 Download 683  
 DPI Scaling 330  
 DPX 233  
 Drag & Drop 214  
 DRM 223

## - E -

E.131 496  
 Edge Blending 69, 88  
 Edge Blending Parameters 88

Edit 444  
 Editions 15  
 Editors 358  
 Effects 128  
 Element Tree 369  
 Elgato Stream Deck 200, 483  
 Elgato Stream Deck mini 200, 483  
 Elgato Stream Deck XL 200, 483  
 Embedded Audio 223  
 Emergency License Reset 33  
 Emergency License Reset 33  
 Encoding Settings 223  
 Encrypt File 223  
 Encryption 223  
 Encryption settings 223  
 Enter Fullscreen 342, 367  
 Epic Games 683  
 Error 578  
 Error log 578  
 Ethernet 496  
 Event 616  
 Examples for a Script 655  
 Execute a Script 655  
 Expiry Date 15, 223  
 Export 223  
 Expression 551  
 External Processing 223  
 Eyefinity 23

## - F -

F1 712  
 Fade In 152  
 Fade Out 152  
 FadeProgValue 619  
 FadeValue 619  
 Favorite Folders 269, 274, 369  
 FBX 238  
 FFD 84, 410  
 FFDs 167  
 FFmpeg 238  
 File formats 711  
 File ID 214, 425, 509  
 File Suffix 711  
 Filter Expressions 619  
 Firewall 691  
 First Project 266  
 Floating Windows 358  
 Focus Tracking 392  
 Folder 214  
 Folder ID 214, 425, 509  
 Font Size 264  
 Force 24-Bit Caching 233  
 Force 48-Bit Caching 233  
 FPS 113, 152  
 Frame Blending 223  
 Framerate for Image Sequence 233  
 Frames per Second 113  
 Free Form Deformers 167  
 Freerun 152  
 Full Screen 444  
 Full Screen on Startup 444  
 Fullscreen 19, 342  
 Fullscreen Renderer 334, 342  
 Fullscreen Renderer and User Interface 330  
 FX 128, 168, 425, 509

## - G -

Generate SMPTE 673  
 generating timecode 680  
 Generative Content 128  
 Generative Textures 254  
 Generic devices 410  
 Geometry 238  
 Geometry Modifier 69, 410  
 Geometry Modifiers 167  
 Geometry Objects 238  
 Getting Started 65  
 GIF 238  
 Global Offset 113  
 Global Position Offset 113, 152  
 Global Position Scale 152  
 Global Scale Offset 113  
 Global Value 185, 468, 542  
 Global Values 413  
 Global Variable 664  
 Gobo 425, 509  
 GoTo 655  
 GPU 23  
 GPU load 367  
 GPU Sync 354  
 Gradient 254  
 Group 277  
 Groups 185, 269, 277, 468, 542



**- H -**

H.264 223  
 Hap 223, 722  
 HAP Alpha 223  
 HAPQ 223  
 HAP-Q 223  
 HAP-Q Alpha 223  
 Hardware 23  
 Hardware lock 15  
 Hardware Requirements 23  
 Hardware usage 367  
 Help 62, 362  
 Hexadecimal 496  
 Host System 181, 464, 509  
 HTML 260, 691  
 HTML5 260  
 HTML Buttons 691  
 HTML Content 128  
 HTML-Content 260  
 hue 421

**- I -**

Icons 128  
 ID 269, 270, 425, 509, 578  
 Image Formats 238  
 Image Sequence 214, 233  
 Image Sequence Dirextory 233  
 Import 214  
 Import Content 213, 282  
 Import Content Diretory 213  
 Import Directory 214  
 Import File 214  
 Import Image Sequence 214  
 Import new Version 280  
 Indexers 619  
 Inpoint 128  
 Input 181, 464, 509, 556, 578  
 Input Cards 246  
 Input DMX-512 496  
 Insert 583  
 Insert Track 122  
 Inspector 88, 104, 122, 149, 167, 168, 282, 392, 413, 444, 551, 583, 664  
 Inspector Mode 367  
 install 26  
 Installation 26

Installer 15, 26, 444  
 Instance 128  
 Interaction 551, 556  
 Interactive 551  
 Interactivity 444  
 Interanal Sound Card 96  
 ioversal 193, 476  
 ioversal AutoCal 688  
 IP Adress 362  
 Iris 425, 509

**- J -**

Javascript 260  
 JPEG 233, 238  
 JumpCount 149  
 JumpToCue 149  
 JumpToTime 149

**- K -**

Key 223  
 Keyboard Shortcuts 697, 712  
 Keyframe 145, 185, 239, 413, 468, 542  
 Kiosk Browser 633, 698  
 KNX 633, 698

**- L -**

Leave Fullscreen 342  
 LED 246  
 Length 113, 152  
 Library 167, 168, 173, 185, 193, 206, 410, 456, 468, 476, 489, 542, 714  
 Library Editor 173, 410, 456  
 License 15, 33, 223, 292  
 License Handling 362  
 License List 223  
 License Menu 362  
 License Menue 33  
 License Model 15  
 License Status 33, 362  
 License Update 33  
 Licenses and Editions 15  
 Lighting Control 496  
 Lighting Desk 425, 509  
 Limitations of Trial Version 32  
 Link 128  
 List of Script Commands 633, 697, 698

Literal Value 583, 595  
 Live 102, 334, 340, 383  
 Live Audio 96, 169, 353  
 Live Input 246  
 Live Video Sources 246  
 Live Volume 96  
 Load 288  
 Load Project 293  
 Load Recent 288  
 Local Project Path 311  
 Local Variable 664  
 Local Variables 655  
 Location 444  
 Lock to time 128  
 locking 412  
 Log 619, 645  
 Loke to track 128  
 LTC 354, 673, 675  
 LUT 423  
 LUTs 423

## - M -

Magewell 246  
 Main Menu 214, 362  
 MainContent 633, 698  
 Manage Content 269  
 MaNet 496  
 MA-Net 181, 464, 509  
 Mapping 84, 167  
 Mask 128  
 Master 20, 292, 293, 306, 307, 311, 321, 322, 362, 583  
 Mathematical Expressions 392  
 Metadata 270  
 Microsoft Office 239  
 Microsoft Powerpoint 193, 239, 476  
 Misc 697  
 miscellaneous 697  
 Mix Level 334, 340  
 Mixing Level 334, 403  
 Mode 551  
 Modifier 84, 377  
 Modulator 595  
 Monitor 578  
 Mosaic 23  
 Move 377  
 MP3 238  
 MPEG 223

Multi Clip 152  
 Multi GPU 23  
 Multi Selection 392  
 Multi-Channel Audio 223  
 Multiple Editors of the same type 358  
 Multiple Projectors 88  
 Multiple Systems 246, 292, 583  
 Multi-System 20, 321  
 Multi-Systemes 306  
 Multi-User 292, 306  
 Mute Live Audio 96  
 Mute Preview Audio 367

## - N -

Name 269, 270  
 NDI 206, 246, 347, 489, 714  
 NDI Output Stream 347  
 NDI Streamer 206, 489, 714  
 Nested Sequence 113  
 Nested Sequence Content 113  
 Nesting 113  
 Network 311  
 Newtec NDI 206, 489, 714  
 Newtek NDI 246, 347  
 Node 559, 563, 570, 578, 583, 604  
 Node Editors 559  
 Node System 559, 563, 570, 578, 583  
 Node System Basics 559  
 Node System Editor 559, 570, 583, 604  
 Node System Monitor 559, 578  
 Node Types 559  
 Normal 145  
 Notch 239  
 Notch Blocks 239  
 Notch Playback 239  
 NotchLC 223  
 Notes 152, 269, 270  
 Notifications 367  
 Nowpointer 383, 405  
 NVidia Mosaic 23  
 NVidia Quadro 23  
 NVMe 23

## - O -

Offline Activation 33  
 Online Account 33  
 Online Activation 33

optimization 57  
 optimize 57  
 Optimized Cached Format 233  
 Optimized File 223  
 Optimized File Format 287  
 Optimized Video 287  
 OSC Sender 633, 698  
 Osprey 246  
 Outpoint 128  
 Output 67, 69, 80, 84, 88, 128, 330, 377, 688  
 Output DMX-512 496  
 Output Position 69  
 Output Reference 69  
 Output Settings 330  
 Output Setup 80, 330  
 Output Space 67, 69  
 Output Streams 347

## - P -

Page 444  
 Parameter 559  
 Parameters 583, 595, 655, 664  
 Partial Backup 292, 322  
 Paste 383  
 Pause 149, 645  
 PC License 33  
 Performance 57, 563  
 Performance Monitor 367  
 Photoshop 246  
 Photoshop Layers 246  
 Pin to Inspector 392  
 Plaback Provider Time 113  
 Playback 102, 106, 152, 166, 307, 334, 340, 354, 403, 509, 633, 673, 680, 698  
 Playback Dashboard 254  
 Playback Editor 106, 145, 149, 152, 166, 340, 383, 583  
 Playback Mixing Engine 102, 334, 340, 673  
 Playback Mode 403  
 Playback Order 106  
 Playback Provider 106, 113, 128, 152  
 Playback Speed 403  
 Playback Sync 20, 292, 307  
 Playbacks 403  
 Playbacks Window 106  
 Playhead 383  
 Playlist 102, 106, 152, 166, 282  
 Plugin for Unreal 683  
 PME 102, 334, 340, 633, 673, 698  
 PME Live 334, 342  
 PME Preview 334  
 PNG 233, 238  
 Poll 563  
 Polling 559, 563, 570  
 Port 80 691  
 Postroll 128  
 Powerpoint 193, 239, 476  
 PowerPoint Remote 193, 476  
 PPT 239  
 PPT Device 633, 698  
 PPT Remote 193, 476  
 pptx 193, 239, 476  
 Preroll 128  
 Presets 223  
 preventtion 412  
 Preview 334  
 Preview 102, 334, 340, 383  
 Preview Audio 96, 169, 353  
 Preview Outputs 80  
 Preview Volume 96  
 principles 9  
 Pro Res 422 722  
 Procedural Content 128, 254  
 Process Device 633, 698  
 Process Polling 563  
 Process Pull 563  
 Process Push 563  
 Programer Mode 145  
 Programmer 392, 405, 413, 633, 698  
 Programmer Mode 367, 392, 405, 413  
 Programmer Values 405  
 ProgValue 619  
 Project auto loading 419  
 Project Explorer 166, 214, 223, 269, 270, 277, 282, 369, 425, 509, 604, 664  
 Project File 223, 288  
 Project Folder 288  
 Project Path 266  
 Project Settings 96, 214, 238, 287, 288, 307  
 Projection Mapping 84, 88  
 Promt 655  
 Property Value 583, 595  
 ProRes 223  
 Protocol 173, 456, 496  
 Provider 128  
 Proxy Encoding 214  
 Proxy File Settings 223  
 Proxy Files 223, 233, 287  
 PSD-File 246

Pull 559, 563, 570  
 Pulsating 570  
 Pulse 559, 563  
 Pulses 570  
 Push 559, 563, 570

## - Q -

quick start guide 9  
 QuickBlend 88  
 Quicktime 223

## - R -

RAM 23  
 Rate 551  
 recommendations 57  
 Reconnect 173, 456  
 Recover File 288  
 Redundant Backup 292, 322  
 Refresh Video Source 246  
 Release Notes 62  
 Releative Keyframe 145  
 remote 26  
 remote deploy 26  
 Remove Output Reference 69  
 Remove Surface from Canvas 69  
 Render Editor 80, 104, 167, 340, 377  
 RenderTraget Size 69  
 Reset Connection 173, 456  
 Reset Outputs 330  
 Reset System to Virtual 293  
 Reset Video Source 246  
 Restart 26  
 RGB 421, 619  
 RGBA 619  
 Role 293, 362  
 Rotate 377  
 Run 444, 645  
 Run (with Promt) 655  
 Run a Script 655  
 Run with Parameters 655

## - S -

sACN 181, 185, 321, 464, 468, 496, 509, 542  
 saturation 421  
 Save 288

Save on Local System 293  
 Save on Master 293  
 Save Project 293  
 Save Project Copy 288  
 Save Project File only 288  
 Scale 377  
 Schedule 616  
 Schedule Editor 616  
 Scheduler 616  
 Screen Scaling 330  
 Script 583, 595, 645, 655, 664  
 Script Code 149  
 Script Command 149, 152, 444, 556, 583, 595, 616, 645  
 Script Commands 173, 444, 456, 618, 619, 633, 664, 684, 698  
 Script Editor 618, 645, 655  
 Script Examples 619, 655, 684  
 Script Monitor 618, 645  
 Script Parameters 655  
 Script Reference 618  
 Script Referenece 633, 698  
 Script Server 684  
 Script Server Authentication 684  
 Script Syntax 618, 619  
 Scripting 181, 322, 464, 509, 551, 618, 619, 664  
 Scripts 645  
 Search 392  
 Selection 604  
 Send DMX 185, 468, 542  
 send timecode 680  
 sending timecode 680  
 sequence 102, 106, 113, 122, 128, 149, 282, 383, 673, 680  
 Session 20, 292, 293, 306, 307, 311, 321, 583  
 Session Management 293  
 Session Member 20, 292, 293, 306, 307, 311, 321, 322, 354, 362, 559, 583  
 Session Mode 20  
 Session Status 362  
 Set DataSync Adapter 293  
 Set Fullscreen Mode 347  
 Set Outputs of a System 330  
 Set Streaming Mode 347  
 Settings 392, 444, 619  
 Shader 254  
 Shaders 168  
 Shaper 425, 509  
 Shared Device Group 277  
 Shortcut 214  
 Shortcuts 113, 383, 697, 712

- shortkeys 392
  - Show Context 645
  - Show Status 645
  - Single Clip 152
  - Size 444
  - Slider 444
  - Sliders 15, 444
  - Slides 239
  - Smart Dongle 33
  - Smart Dongle License 33
  - SMPTE 321, 354, 673
  - SMPTE Chase Lock 676
  - SMPTE Clip 128, 673
  - SMPTE clip container 680
  - SMPTE Input 673, 676
  - SMPTE IO 673
  - SMPTE IO Interface 675
  - SMPTE out 680
  - SMPTE Out Channel 680
  - SMPTE Output 673
  - SMPTE Output Mode 680
  - Softedge 88
  - Softedge Blending 69, 88
  - Softedge Projection 88
  - Solid Color 254
  - Source 551, 556
  - SourceA 551
  - Spout 246
  - Standard 392
  - Standard Audio 96, 169, 354
  - Standard Mode 367, 392
  - Start Channel 181, 464, 509
  - Start Vertex Service 26
  - Startup 266
  - Startup Page 266, 293, 444
  - Status Bar 358, 367, 413
  - Status LED 507
  - Stop 149
  - Stop Vertex Service 26
  - Storage 23
  - Stream Deck 200, 483
  - Stream Deck mini 200, 483
  - Stream Deck XL 200, 483
  - Stream Identifier 347
  - Streamdeck 200, 483, 633, 698
  - Streamlabs 246
  - Structure 619
  - Sub-Composition 113, 604
  - Subnet 496
  - Surface 67, 69, 80, 84, 88, 102, 104, 128, 168, 330, 347, 377, 509, 688
  - Swap to Live on Mix 334
  - Switch 145
  - Switch between Value and Programmer Mode 413
  - sync 676
  - Sync Clock 96, 307, 354, 673, 676
  - Sync Clock Source 287
  - Sync Clock Source System 307
  - Sync Proxy files only. 311
  - synch 676
  - synchronize 676
  - Syntax 618
  - System 80, 88, 169, 246, 293, 322, 347, 354, 377, 444, 509, 559, 563, 583, 633, 673, 676, 698
  - System Basics 14
  - System Clock 307, 354, 362
  - System License 33
  - System Output Setup 69, 330
  - System Role 362
  - System Status 362
  - System View 377
  - Systems 369
  - Systems Manager 633, 698
- T -**
- Tag 270, 655
  - Tags 269, 270
  - Target 128, 551, 583
  - Target System 583
  - Target Systems 269, 311, 583
  - Targets for Content Items 269
  - TCP 321, 684, 691
  - TCP Client 633, 698
  - TCP Connection 633, 698
  - Template 166
  - Template Clip 122, 166
  - Template Name 166
  - temporary pre-load 676
  - TempPreload 676
  - Test-Pattern 254
  - Text 264
  - Text Container 264
  - Text Content 264
  - Texture 168
  - Texture Size 264
  - Textures 254
  - TGA 233
  - Throttle 563

Throttling 563  
 TIFF 233, 238  
 Timecode 354, 383, 673, 675  
 Timeline 113  
 Timestamp 578  
 Tool Tip Monitoring 570  
 Top Bar 362  
 Top Left Coordinates 287  
 Touch Interface 444  
 Touchscreen Interfaces 15, 444  
 Track 113, 122, 128, 152, 383  
 Track Clip Mode 152  
 Track Settings 122  
 Transcode 223, 722  
 Transcoder 722  
 Transcoding 223, 722  
 Transfer 311  
 transmitting timecode 680  
 Transport Mode 152  
 Tray Icon 26  
 Trial 26, 32  
 Trial Version 32  
 Trigger 152, 181, 464, 509, 551, 556  
 Trigger by date 616  
 trigger Editor 181, 464, 509, 551  
 Triggering 509, 556

## - U -

UDP 321  
 UDP Sender 633, 698  
 UI 383  
 UI on Primary 330  
 unintentional changes 412  
 Universe 181, 464, 496, 507, 509  
 Universe Numbering 496  
 Unlink 128  
 Unreal Engine 683  
 Update Playlist 282  
 Update Sequence 282  
 Use Track Channel Map 169  
 User Account 683  
 User Color 122, 269, 270  
 User Interface 19, 358, 369, 383  
 User Interface and Fullscreen Renderer 330  
 User Properties 269, 270  
 UV-Pattern 254

## - V -

Validate 595  
 Validate Inputs 595  
 Validate This Node 595  
 Validation 595  
 Validity 15  
 Value 619  
 Value Mode 145, 367, 392, 405, 413  
 Variable 618, 664  
 Versioning 223, 269, 280  
 Vertex Autostart on Windows 419  
 Vertex background Service 293  
 Vertex Background Services 26  
 VERTEX Data Formats and File Suffixes 697  
 Vertex Edit 15  
 Vertex NDI Streamer 206, 489, 714  
 Vertex Play 15  
 Vertex Project 306  
 Vertex Service 26  
 Vertex Suite 15  
 VERTEX Touch 15, 444  
 VERTEX Vioso 688  
 Vertices 84, 167, 377  
 Video 223  
 Video Capture Cards 246  
 Video Codecs 223  
 Video Containers 223  
 Video Encryption 223  
 Video FX 168, 410  
 View 340  
 Views 358  
 Vioso Blend File 84  
 Vioso 69, 88, 688  
 Vioso AutoCal 84, 88, 688  
 Vioso Blend File 88  
 Vioso License 688  
 Vioso Warp File 84, 88  
 Virtual Machines 23  
 Virtual System 293  
 Visualize Pulses 570  
 Volume 169  
 vxdi 173, 410, 456  
 vxdx 173, 410, 456

## - W -

Wait 149

WaitTime 149  
warp file 69, 84, 88, 688  
Warping 69, 84, 167  
Watchfolder 166, 214, 269, 282, 419  
WAV 238  
Webbrowser Content 633, 698  
Webcam 246  
Webcontrol 691  
Webserver 691  
Weiterführende Info 246  
Wheel 392  
white balance 421  
Window 358  
Window Docking 358  
Window Undocking 358  
Windows 10 23  
Windows 10 Desktop Setting 330  
Windows 8.1 23  
Windows Autostart 419  
Windows Startup 419  
Windows Volume 96, 169  
Wiring 181, 392, 464, 509, 551  
Wiring Editor 181, 464, 509, 551  
Workflow 67, 559  
Workspace 340, 358